

Modelo lineal	Gh11yGh12Lineal.mdl
PI sin cambio de parámetros	Pisimple.slx
PI con cambio de parámetros	controlp1p2if2.mdl
Aplicación del control feedforward sobre el modelo	Feedforwardcomp.slx
Aplicación del control LQR sobre el modelo	LQR01.m lqr04.mdl
Base de datos	Proyecto_MatrizRandSaturDEF.m proyecto03pto4.slx
Método de mínimo índice de desempeño	Proy_Script_Predict_min.m buscaptocercano3NminNmax.m calculoIDs.m proyectopredictivo01.slx
Método de índices ponderados	Proy_Script_Predict_abcd.m buscaptocercano3NminNmax.m calculoIDs.m proyectopredictivo01.slx
Método mediante uso de programación lineal	CPredictivoDEF.m buscaptocercano5N200.m calculoIDs.m funlinprog2.m proyectopredictivo01.slx Matrizmadrepto1N1000Sat.mat
1 trayectoria de ejemplo	CPredictivo1trayectoriaDEF.m buscaptocercano5N200.m calculoIDs.m funlinprog2.m proyectopredictivo01.slx Matrizmadrepto1N1000Sat.mat
100 aleatorias valores medios	CPredictivoDEF.m
Estimación de M óptimo	MoptimoDEF.m
Aplicación al concurso	predictivoconcurso.m buscaptocercano5N200.m calculoIDs.m funlinprog2.m proyectopredictivo01.slx Matrizmadrepto1N1000Sat.mat Matrizmadrepto4N1000Sat.mat Matrizmadrepto5N1000Sat.mat Matrizmadrepto6N1000Sat.mat

Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías Industriales

Control Predictivo Basado en Datos. Aplicación a un Sistema de Cuatro Tanques.

Autor: Vicente Simón Gómez

Tutor: David Muñoz de la Peña Sequeda

Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Control Predictivo Basado en Datos. Aplicación a un Sistema de Cuatro Tanques.

Autor:

Vicente Simón Gómez

Tutor:

David Muñoz de la Peña Sequeda

Profesor titular

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2015

Agradecimientos

Mostrar mi agradecimiento al tutor David Muñoz por el tema facilitado y la ayuda prestada.

Resumen

En este proyecto se desarrolla la implementación de un controlador predictivo basado en datos (DbPC) que permite resolver el problema de seguimiento de un único punto objetivo sobre una planta formada por cuatro tanques. A diferencia de otras estrategias de control, ésta no usa un modelo, sino una base de datos para abordar el problema de seguimiento. En esta base de datos se almacena un histórico de las trayectorias llevadas a cabo por la planta de los cuatro tanques en bucle cerrado usando diferentes controladores. Con vistas al análisis y a la validación del método empleado se aplican diferentes simulaciones sobre la planta de cuatro tanques y éstas se comparan con los resultados obtenidos por las diferentes estrategias de control que se han utilizado para generar la base de datos.

Abstract

This paper presents the implementation of a data-based predictive control (DbPC) that allows the trajectory control problem of a single set-point of four interconnected water tanks to be solved. The main difference of this method control is the use of an algorithm designed from the outputs and inputs of the simulation code in which no explicit model exists to tackle the problem. The data in this database is stored as historical trajectories carried out through the quadruple-tank process in closed-loop control using different control strategies. This is verified by a comparative study simulation between the DbPC and different control strategies that have been used to generate the database.

ÍNDICE

Agradecimientos	i
Resumen	iii
Abstract	v
Índice	vi-vii
Índice de Figuras	viii-ix
Índice de Tablas	x
1. Introducción	1-2
1.1 Objetivos del proyecto	1
1.2 Definición del proyecto	2
2. Modelo de la planta de los cuatro tanques	3-8
2.1 Descripción de la planta de los cuatro tanques	3-4
2.2 Cálculo del modelo del sistema	5-8
2.2.1 Modelo no lineal	5-6
2.2.2 Modelo lineal	6-8
3. Estrategias de control estudiadas sobre el modelo de la planta de los cuatro tanques para crear una base de datos	9-25
3.1 Resumen de las diferentes estrategias	9
3.2 Control de procesos multivariables	10-11
3.3 Medida de las interacciones. Método de Bristol	11-12
3.4 Control proporcional (P)	12-13
3.5 Control proporcional integral (PI) por cancelación de dinámica	14-18
3.5.1 PI sin cambio de parámetros	14-15
3.5.2 PI con cambio de parámetros	16-18
3.6 Control anticipativo (Feedforward)	18-21
3.6.1 Principios del control anticipativo	18-19
3.6.2 Aplicación del control feedforward sobre el modelo	19-21
3.7 Regulador óptimo cuadrático LQR	21-25
3.7.1 Introducción al control óptimo LQR	21-23
3.7.2 Aplicación del control LQR sobre el modelo	23-25
4. Base de datos	26-28
4.1 Composición de la base de datos	26-27
4.2 Métodos de cálculo de los estados iniciales	27-30

5. Descripción del controlador predictivo basado en datos (DbPC)	31-45
5.1 Estado del arte	31-34
5.1.1 Control predictivo basado en modelo (MPC)	31
5.1.1.1 Perspectiva histórica	31-32
5.1.1.2 Conceptos básicos del MPC	33-34
5.2 Diferencias básicas entre MPC y DbPC	34-36
5.3 Algoritmos de resolución	36-45
5.3.1 Método de mínimo índice de desempeño	36-39
5.3.2 Método de índices ponderados	40-40
5.3.3 Método mediante uso de programación lineal	40-45
6. Análisis/validación del controlador	46-54
6.1 Comparativa de los índices de desempeño	46-51
6.1.1 1 trayectoria de ejemplo	46-50
6.1.2 100 aleatorias valores medios	50-51
6.2 Optimización de la base de datos	51-54
6.2.1 Estimación de M óptimo	51-52
6.2.2 Coste medio/tiempo de cálculo medio/tamaño de base de datos	53-54
7. Aplicación al concurso	55-57
Referencias	58

Índice de figura

Figura 2.1. Esquema de la planta de Johansson	3
Figura 2.2. Esquema de simulink del modelo no lineal de la planta de los cuatro tanques	5
Figura 2.3. Modelo lineal del sistema de la planta de los cuatro tanques	6
Figura 2.4. Esquema de simulink del modelo lineal de la planta de los cuatro tanques	8
Figura 3.1. Estudio de la dependencia de entradas y salidas de nuestro esquema de control	10
Figura 3.2. Esquema de simulink del controlador proporcional de la planta de los cuatro tanques realimentado	13
Figura 3.3. Representación de las salidas h_1 y h_2 de la planta de los 4 tanques controlado por un controlador proporcional en bucle cerrado	13
Figura 3.4. Modelo de simulink del control PI donde h_1 depende de q_b y h_2 de q_a .	15
Figura 3.5. Comparativa de h_1 y h_2 usando o no los emparejamientos recomendados del método de Bristol	15
Figura 3.6. Problema de seguimiento	17
Figura 3.7. Esquema de simulink de la implementación de dos conjuntos de controladores PI	17
Figura 3.8. Comparativa de h_1 y h_2 usando G_{C1} y G_{C2} con y sin cambio de parámetros	18
Figura 3.9. Sistema con perturbación a la salida controlado con un controlador anticipativo con control realimentado	19
Figura 3.10. Esquema de simulink del control feedforward de la planta de los cuatro tanques realimentado	20
Figura 3.11. Comparativa de las salidas h_1 , h_2 de la planta de los cuatro tanques controlado por un controlador feedforward y un controlador integrativo en bucle cerrado	21
Figura 3.12. Esquema de simulink del esquema de control del controlador LQR en bucle cerrado	24
Figura 3.13. Representación de las salidas h_1 , h_2 , h_3 y h_4 de la planta de los cuatro tanques controlado por un controlador LQR en bucle cerrado	25
Figura 4.1. Estados iniciales en la base de datos	29
Figura 4.2. Modelo de simulink utilizado para la simulación de las diferentes trayectorias que se almacenan en la base de datos	30
Figura 5.1. Esquema de control de controlador predictivo	33

Figura 5.2. a) Esquema de MPC b) Esquema de control predictivo basado en datos	34
Figura 5.3. Problema de seguimiento	35
Figura 5.4. Esquema de control predictivo basado en base de datos	36
Figura 5.5. Método de aproximaciones	38
Figura 5.6. Método de la bisectriz	38
Figura 5.7. Algoritmo de búsqueda del mínimo ID	38
Figura 5.8. Diferentes modos de búsqueda de estados iniciales	39
Figura 5.9. Esquema del control predictivo basado en base de datos con método de mínimo ID	39
Figura 5.10. Esquema de control predictivo basado en base de datos usando programación lineal	41
Figura 5.11. Estudio de la factibilidad del problema de optimización	42
Figura 5.12. Método de aproximaciones	43
Figura 5.13. Método de la bisectriz	44
Figura 5.14. Esquema de control predictivo basado en base de datos método usando programación lineal optimizado	44
Figura 5.15. Variables de decisión del controlador predictivo por programación lineal	45
Figura 6.1. Comparativa del valor de los actuadores q_a y q_b de los diferentes controladores sobre una trayectoria	46
Figura 6.2. Comparativa de alturas de los diferentes controladores sobre una trayectoria	47
Figura 6.3. Comparativa del índice de desempeño de los controladores estudiados sobre una trayectoria	48
Figura 6.4. Comparativa del valor de los actuadores q_a y q_b de los diferentes controladores sobre una trayectoria	49
Figura 6.5. Comparativa de alturas de los diferentes controladores predictivos sobre una trayectoria	50
Figura 6.6. Comparativa de ID de los diferentes controladores en 100 conjunto de trayectorias partiendo de h_0 aleatorias	51
Figura 6.7. Representación de la suma de ID de un conjunto de trayectorias usando bases de datos de diferente tamaño	52
Figura 6.8. Representación del coste computacional de una simulación con bases de datos con diferente tamaño	53
Figura 7.1. Representación de las alturas del concurso de la CEA usando un controlador DbPC	55
Figura 7.2. Comparativa de las alturas del concurso de la CEA usando un controlador DbPC, P, PI y LQR	56

Figura 7.3. Comparativa de los índices de desempeño del concurso de la CEA usando un controlador DbPC, P, PI y LQR	57
--	----

Índice de tablas

Tabla 3.1. Resumen de los diferentes controladores implementados sobre la planta de los cuatro tanques	9
Tabla 4.1. Composición de la base de datos	27

1. Introducción

1.1 Objetivo del proyecto

Se nos presenta el objetivo de resolver de un problema de control para un sistema del cual no es posible obtener un modelo aproximado o preciso debido a que dicho sistema presenta una dinámica difícil de identificar. La solución que se presenta es la obtención de un controlador predictivo basado en una base de datos (DbPC) de dicho sistema en lugar del modelo del mismo. En esta base de datos estaría recogido el comportamiento de dicho sistema a lo largo de un periodo de tiempo t suficientemente grande que nos ayude a predecir los valores de los actuadores sin necesidad de un modelo. Como paso previo a la implementación del control predictivo basado en datos sobre un sistema desconocido se opta por validar el algoritmo sobre un sistema cuyo modelo haya sido previamente estudiado y sea conocido.

Como primer paso para saber si DbPC ser extrapolable o no a un sistema cuya dinámica se desconozca en este proyecto se aborda exclusivamente el problema de seguimiento para alcanzar un único punto objetivo haciendo uso del DbPC sobre el modelo de los cuatro tanques. Así pues para aplicar nuestro DbPC se establece que en cada base de datos se encuentren almacenados todas las trayectorias que convergen a un único punto objetivo de tal forma que para alcanzar i puntos de referencias harían falta tener i bases de datos. En una ampliación del trabajo habría que resolver el problema de seguimiento para cualquier alcanzar cualquier punto objetivo con el uso de una base de datos que recoja el conjunto de puntos objetivo.

1.2 Definición del proyecto

Los pasos que se han llevado a cabo para alcanzar nuestros objetivos se puede dividir en tres bloques:

1. Estudio de la planta de los cuatro tanques e implementación de diferentes estrategias de control
2. Creación de una base de datos
3. Control predictivo basado en datos

En primer lugar, se obtiene a partir de las ecuaciones diferenciales del modelo de la planta de los 4 tanques el modelo no lineal del sistema. A continuación, se calcula el modelo lineal entorno a un punto de funcionamiento como base de partida del estudio de controladores en bucle cerrado. Luego, se aplican diferentes estrategias de control, primero sobre el modelo lineal y más tarde sobre el modelo no lineal, que resolverán el problema de seguimiento en bucle cerrado para alcanzar un único punto objetivo.

En segundo lugar, se genera una base de datos estática donde se almacena un histórico de las trayectorias llevadas a cabo por la planta de los cuatro tanques en bucle cerrado con los diferentes controladores durante un periodo de tiempo t . Estas trayectorias son generadas haciendo uso del modelo no lineal del sistema. Este paso intenta emular lo que en la práctica se trataría de datos que se han ido almacenando a lo largo de los años en la planta real.

En tercer lugar, se programa un control predictivo basado en datos (DbPC) que nos permite resolver el problema de seguimiento a partir de los datos proporcionados por trayectorias almacenadas en la base de datos ya generada. En este bloque se detalla cómo se ha llegado a obtener dicho controlador y se describen los puntos más importantes del algoritmo utilizado.

Finalmente, una vez programado el DbPC se pasa a validar y analizar dicho controlador comparándolo con las diferentes estrategias de control que han servido como partida de la base de datos. Por último, se aplica el DbPC al concurso de la CEA como base para comprobar las prestaciones de dicho controlador.

2 Modelo de los cuatro tanques

2.1 Descripción de la planta de los cuatro tanques [CEA]

La planta de los cuatro tanques es una implementación de la planta propuesta por Karl H. Johansson en el artículo titulado “*The quadruple-tank process*”, publicado en la revista *IEEE Transactions on Control Systems Technology*, vol 8 (2000). Esta planta ha sido ampliamente usada como planta de laboratorio debido a su interés como problema de control y su sencillez de construcción y uso.

En la figura 1.1, que se muestra a continuación, se ilustra el esquema de la planta de Johansson. Ésta consta de cuatro depósitos, dos inferiores (tanques 1 y 2) y dos superiores (tanques 3 y 4) que desaguan en los tanques inferiores. Los tanques se llenan con dos bombas que impulsan sendos caudales q_a y q_b desde el depósito colector situado en la parte baja de la planta. Estos caudales entran en sendas válvulas de tres vías que dividen el caudal entre dos ramas, de forma que una fracción (denominadas γ_a y γ_b) se envía por una de las ramas y el resto por la otra. Las fracciones γ_a y γ_b se indican en tanto por uno y se fijan manualmente mediante la apertura de dichas válvulas.

De esta forma, el caudal $\gamma_a q_a$ entra en el tanque 1 y el caudal $(1-\gamma_a)q_a$ entrará en el tanque 4. Del mismo modo, el caudal $\gamma_b q_b$ entra en el tanque 2 y el caudal $(1-\gamma_b)q_b$ entrará en el tanque 3. El tanque 3 se descarga sobre el tanque 1 y este sobre el depósito colector. El tanque 4 se descarga sobre el 2 y este en el tanque colector.

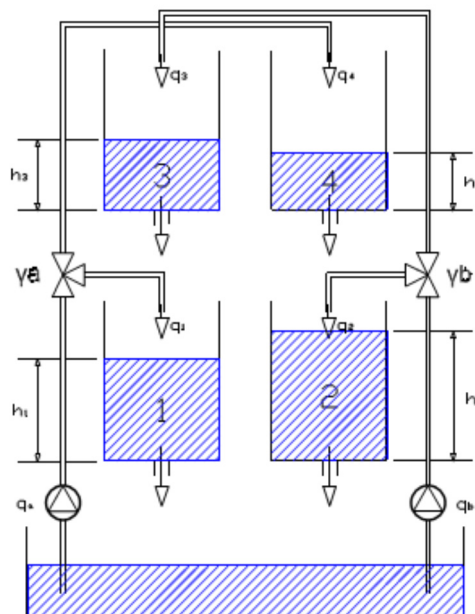


Fig 1.1. Esquema de la planta de Johansson

Esta planta es multivariable pues presenta dos variables manipulables q_a y q_b . Típicamente, las variables que se desean controlar son el nivel de líquido en los depósitos inferiores. La dinámica de esta planta puede presentar ceros de transmisión según los valores de la apertura de las válvulas de tres vías, γ_a y γ_b . Asimismo, la planta presenta una dinámica no lineal y se describe bien mediante un modelo de orden cuatro basado en primeros principios. Además la planta presenta restricciones en las actuaciones (caudales máximos y mínimos) y en las variables del proceso (niveles máximos y mínimos en los depósitos).

Modelo de la planta de los cuatro tanques

El modelo de la planta de Johansson se puede derivar de primeros principios. Este modelo viene dado por las siguientes ecuaciones diferenciales

$$\begin{aligned}
 A \frac{dh_1}{dt} &= -a_1 \sqrt{2gh_1} + a_3 \sqrt{2gh_3} + \gamma_a \frac{q_a}{3600} \\
 A \frac{dh_2}{dt} &= -a_2 \sqrt{2gh_2} + a_4 \sqrt{2gh_4} + \gamma_b \frac{q_b}{3600} \\
 A \frac{dh_3}{dt} &= -a_3 \sqrt{2gh_3} + (1 - \gamma_b) \frac{q_b}{3600} \\
 A \frac{dh_4}{dt} &= -a_4 \sqrt{2gh_4} + (1 - \gamma_a) \frac{q_a}{3600}
 \end{aligned} \tag{2.1}$$

en las h_i que denota el nivel de líquido en el tanque i . Este nivel se mide en metros. Los caudales de cada rama medidos en metros cúbicos por hora se denotan q_a y q_b y son variables manipulables.

El parámetro a_i es la sección equivalente del orificio de descarga del depósito i y la sección de todos los tanques es igual y se denota por A . Las secciones se miden en metros cuadrados. Los parámetros γ_a y γ_b indican la apertura de la válvula de tres vías. La unidad de tiempo del modelo es segundos.

Como se explicó anteriormente, la planta real presenta dinámicas inmodeladas en los caudales de alimentación de los tanques así como incertidumbres y ruidos en las medidas que no están presentes en el modelo de primeros principios. Además la actualización del valor de las variables manipulables que se aplicarán y la actualización de las medidas de los sensores se realiza cada cinco segundos.

2.2 Cálculo del modelo del sistema

2.2.1 Modelo no lineal

A partir de las ecuaciones diferenciales que se nos presenta en el modelo de la planta de Johansson se crea un modelo no lineal haciendo uso del Álgebra de Bloques, que permite operar con sistemas dinámico de forma algebraica.

$$\begin{aligned}
 A \frac{dh_1}{dt} &= -a_1 \sqrt{2gh_1} + a_3 \sqrt{2gh_3} + \gamma_a \frac{q_a}{3600} \\
 A \frac{dh_2}{dt} &= -a_2 \sqrt{2gh_2} + a_4 \sqrt{2gh_4} + \gamma_b \frac{q_b}{3600} \\
 A \frac{dh_3}{dt} &= -a_3 \sqrt{2gh_3} + (1 - \gamma_b) \frac{q_b}{3600} \\
 A \frac{dh_4}{dt} &= -a_4 \sqrt{2gh_4} + (1 - \gamma_a) \frac{q_a}{3600}
 \end{aligned}
 \tag{2.1}$$

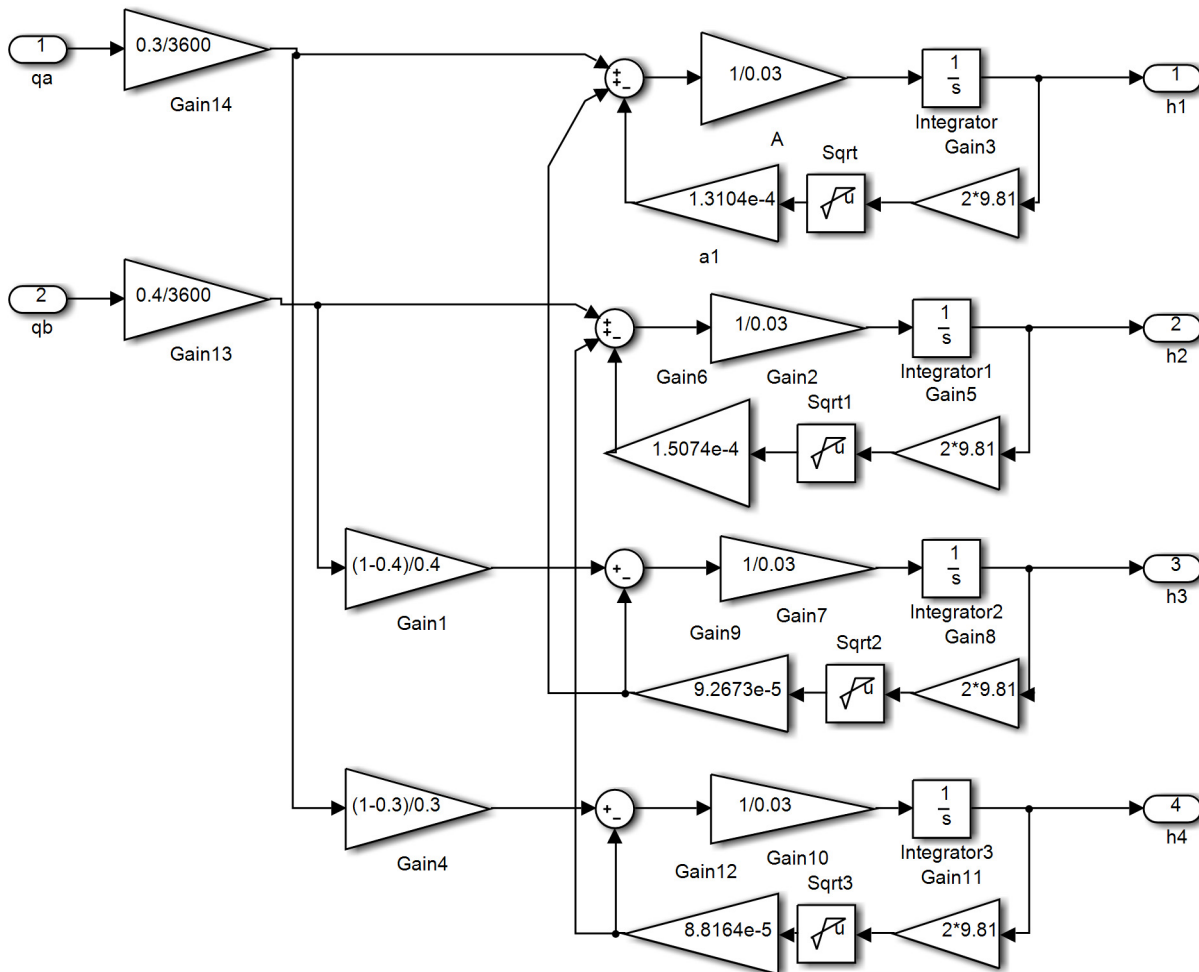


Fig 1.2. Esquema de simulink del modelo no lineal de la planta de los cuatro tanques

Cabe señalar que cuando simulemos las diferentes estrategias de control sobre el modelo no lineal en bucle cerrado debemos saturar tanto las entradas como las salidas, ya que tanto las válvulas como los depósitos tienen unas restricciones de operación.

$$0.2m \leq h_i \leq 1.2m \quad \forall i = 1, 2, 3, 4 \quad (2.2)$$

$$0m^3/h \leq q_z \leq 2.5m^3/h \quad \forall z = a, b$$

2.2.2 Modelo lineal

Una vez hemos obtenido el modelo no lineal calculamos el modelo lineal entorno a un punto de funcionamiento. Para ello calculamos las funciones de transferencia que relacionan las entradas q_a y q_b con las salidas h_1 y h_2 por medio de la respuesta en escalón.

$$G = \begin{pmatrix} G_{H11} & G_{H12} \\ G_{H21} & G_{H22} \\ 0 & G_{H3} \\ G_{H4} & 0 \end{pmatrix} \quad (2.3)$$

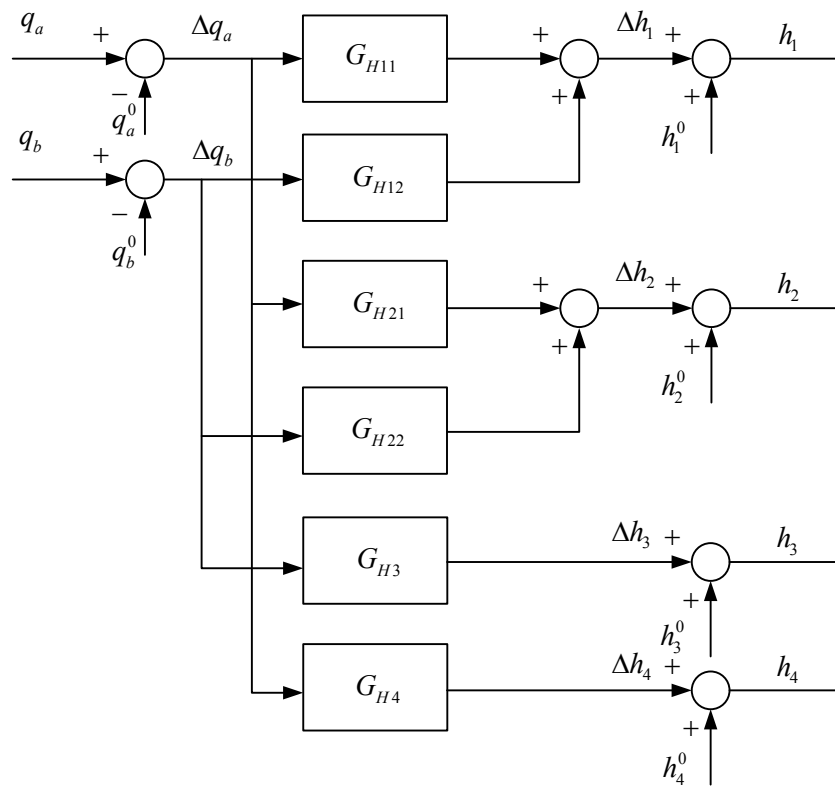


Fig 1.3. Modelo lineal del sistema de la planta de los cuatro tanques

Al tratarse de un sistema multivariable con el objetivo de obtener G pasamos a identificar un modelo de primer orden que relacione cada una de las entradas con cada una de las salidas. De esta forma llegaremos a un modelo de primer orden de la forma:

$$G_H = \frac{K_H}{1 + \tau_H s} \quad (2.4)$$

Se procede de la siguiente forma. Aplicamos un escalón sobre una de las entrada q haciendo que permanezca la otra entrada constante y obtenemos K y τ a partir de la evolución de h_1 , h_2 , h_3 y h_4 .

1. Para calcular G_{H11} aplicamos un escalón sobre q_a : $q_a \neq cte$ y $q_b = cte$

$$G_{H11} = \frac{K_{H11}}{1 + \tau_{H11} s} \quad (2.5)$$

donde K_H se obtiene observando el régimen permanente de la salida respecto la entrada

$$K_{ij} = \frac{\Delta y_i}{\Delta u_j} \bigg|_{u_k cte} \quad \forall k \neq j \quad (2.6)$$

$$K_{H11} = \frac{K_{1.05} + K_{0.95}}{2} \quad (2.7)$$

$$K_{1.05} = \frac{\Delta y}{\Delta u} = \frac{\Delta h_1}{\Delta q_a} \quad (2.8)$$

$$K_{0.95} = \frac{\Delta y}{\Delta u} = \frac{\Delta h_1}{\Delta q_a} \quad (2.9)$$

mientras que τ_H se calcula estudiando el régimen transitorio. Esta constante de tiempo se observa en la gráfica de h a partir del cálculo previo del 63% del incremento final de la salida sumado el valor previo al incremento.

$$\tau_{H11} = \frac{\tau_{1.05} + \tau_{0.95}}{2} \quad (2.7)$$

donde $\tau_{1.05} \rightarrow 0.63\Delta h_1 + h_1^0$ (2.8)

$$\tau_{0.95} \rightarrow 0.63\Delta h_1 + h_1^0 \quad (2.9)$$

De la misma forma se procede para el cálculo de del resto de componentes de G

2. Para calcular G_{H12} aplicamos un escalón sobre q_b : $q_a = cte$ y $q_b \neq cte$ y observamos la evolución de h_1

3. Para calcular G_{H21} aplicamos un escalón sobre q_a : $q_a \neq cte$ y $q_b = cte$ y observamos la evolución de h_2
4. Para calcular G_{H22} aplicamos un escalón sobre q_b : $q_a = cte$ y $q_b \neq cte$ y observamos la evolución de h_2
5. Para calcular G_{H3} aplicamos un escalón sobre q_b : $q_a = cte$ y $q_b \neq cte$ y observamos la evolución de h_3
6. Para calcular G_{H4} aplicamos un escalón sobre q_a : $q_a \neq cte$ y $q_b = cte$ y observamos la evolución de h_4

Si calculamos los modelos de primer orden en torno al punto de trabajo

$$\begin{aligned}
 h_1 &= 0.5955 \\
 h_2 &= 0.6616 & q_a &= 1.7602 \\
 h_3 &= 0.5384 & q_b &= 1.8072 \\
 h_4 &= 0.7682
 \end{aligned}$$

Llegamos a la expresión

$$G_H = \begin{pmatrix} \frac{0.2216}{1+79.2s} & \frac{0.4427}{1+199s} \\ \frac{0.4733}{1+218.5s} & \frac{0.2706}{1+72.65s} \\ \frac{0.596}{1+107s} & 0 \\ 0 & \frac{0.872}{1+202.5s} \end{pmatrix} \quad (2.10)$$

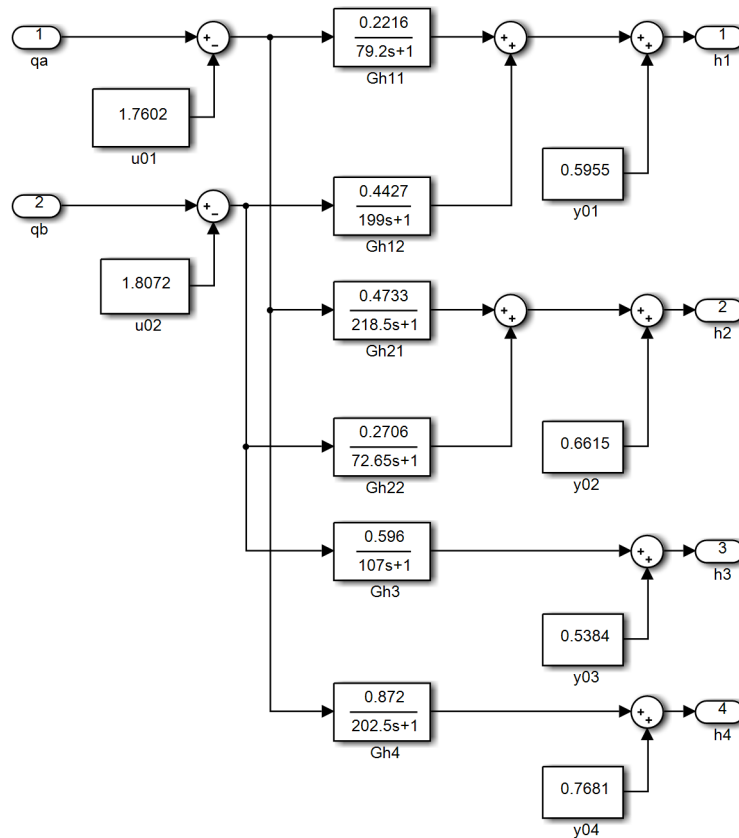


Fig 1.4. Esquema de simulink del modelo lineal de la planta de los cuatro tanques

3 Aplicación de controladores al sistema

3.1 Resumen de las diferentes estrategias de control

En este apartado se trata de describir las diferentes estrategias de control que se han aplicado sobre el modelo de la planta de los 4 tanques en bucle cerrado con el fin de crear a posteriori una base de datos. Los controladores que se han aplicado son:

1. Control proporcional (P)
2. Control proporcional integral (PI)
 - 2.2.1 PI (Sin cambio de parámetros)
 - 2.2.2 PI (Con cambio de parámetros)
3. Control anticipativo (feedforward)
4. Sistema regulador óptimo cuadrático, LQR (linear-quadratic regulator)

Tabla 3.1. Resumen de los diferentes controladores implementados sobre la planta de los cuatro tanques

Controlador	Emparejamientos	Base de datos
Proporcional	$h_1 \rightarrow q_a$ $h_2 \rightarrow q_b$	<i>Si</i>
PI (sin cambio de parámetros)	$h_1 \rightarrow q_b$ $h_2 \rightarrow q_a$	<i>Si</i>
PI (con cambio de parámetros)	$h_1 \rightarrow q_b$ $h_2 \rightarrow q_a$	<i>No</i>
Feedforward	$h_1 \rightarrow q_a$ $h_2 \rightarrow q_b$	<i>No</i>
LQR	$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} \rightarrow \begin{pmatrix} q_a \\ q_b \end{pmatrix}$	<i>Si</i>

3.2 Control de procesos multivariables [1, PP 185-188]

Antes de pasar a la descripción de cada uno de los controladores debemos saber que se trata del control de procesos multivariables. Si utilizamos técnicas SISO en estos sistemas puede llevar a rendimientos muy pobres fundamentalmente por las interacciones existentes entre las diversas entradas y salidas. Esto es así porque una entrada puede afectar a más de una salida y porque una salida puede depender de más de una entrada.

El mejor método para abordar los problemas de control en sistemas multivariables comienza por evaluar las interacciones entre entradas y salidas a fin de poder establecer los mejores emparejamientos posibles entre entradas y salidas. Como se verá, al usar controladores SISO en sistemas multivariables, se obtienen mejores rendimientos si las entradas se emparejan con las salidas correctas

El problema de la dependencia de las entradas y salidas se reduce a la relación de las salidas h_1 y h_2 con las entradas q_a y q_b , ya que h_3 queda definida por q_b y h_4 por q_a .

Así pues si extraen estas variables h_3 y h_4 de nuestro problema de control, nos queda:

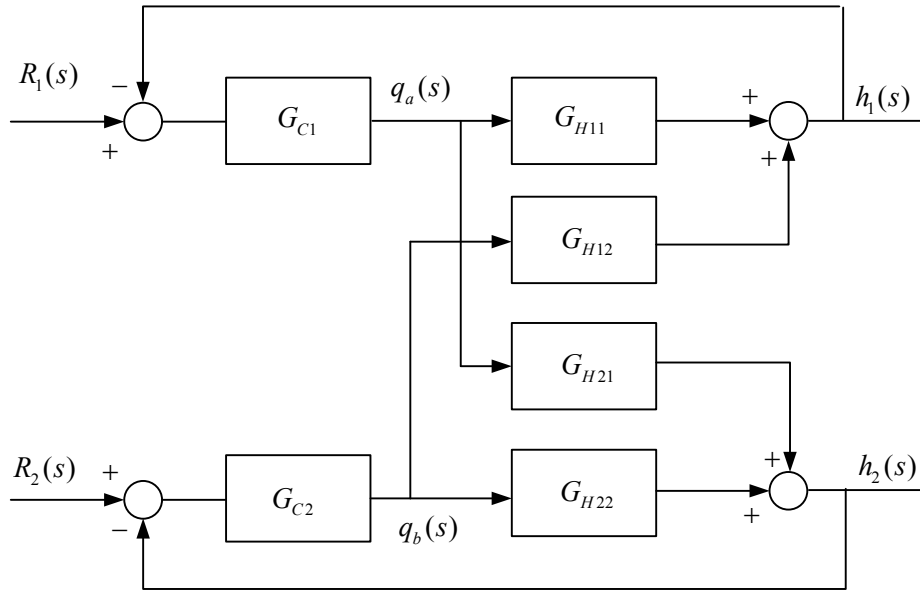


Fig 3.1. Estudio de la dependencia de entradas y salidas de nuestro esquema de control

El efecto cualitativo de las interacciones se puede analizar si suponemos que se inyecta una perturbación en el lazo 1. En este caso el error que introduce la perturbación trataría de ser corregido por el controlador G_{C1} actuando sobre q_a . Los cambios en q_a también afectarían a h_2 a través de la función de transferencia G_{H21} , provocando una desviación e_2 . El controlador G_{C2} trataría de contrarrestar este error e_2 modificando el valor de q_b . El cambio de q_b afecta también a h_1 a través de G_{H12} , por lo que la perturbación en el lazo 1 no sólo afecta al lazo 2, sino que además rebota de nuevo afectando al lazo 1, a través de la perturbación inducida en el lazo 2.

La interacción anteriormente citada provoca que sea difícil sintonizar los controladores. Por ello se concluye que sólo sintonizando simultáneamente los dos controladores se puede garantizar un comportamiento aceptable en bucle cerrado.

3.3 Medida de las interacciones. Método de Bristol [1, PP 188-192]

Para determinar las interacciones de las salidas h_1 y h_2 con las entradas q_a y q_b hacemos uso del método de Bristol de las ganancias relativas. Este método es una técnica que permite evaluar con facilidad las interacciones en régimen permanente y que, en consecuencia, se utiliza para afrontar el problema del emparejamiento entre entradas y

salidas en un sistema multivariable. La idea es que este método nos va a dar una medida del grado en que cada una de las entradas influye sobre cada una de las salidas. Es evidente que para controlar una salida se deberá escoger como entrada aquella que tenga una mayor influencia en su comportamiento.

Si nos centramos en el apartado práctico del método hay que calcular las ganancias estáticas de bucle abierto $K_{ij} \forall i=1,2 \ j=a,b$ ya calculadas en el punto dos.

$$K_{ij} = \left. \frac{\Delta y_i}{\Delta u_j} \right|_{u_k cte} \quad \forall k \neq j \quad (3.1)$$

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \quad (3.2)$$

Así como $K_{Cij} \forall i=1,2 \ j=a,b$ que miden la influencia en régimen permanente de la entrada j sobre la salida i cuando el resto de las variables de salida permanecen en su valor deseado y constantes.

$$K_{Cij} = \left. \frac{\Delta y_i}{\Delta u_j} \right|_{y_k cte} \quad \forall k \neq i \quad (3.3)$$

Llegando a que

$$K_{C11} = \left. \frac{\Delta h_1}{\Delta q_a} \right|_{h_2 cte} = \frac{K_{11}K_{22} - K_{12}K_{21}}{K_{22}} \quad (3.4)$$

$$K_{C12} = \left. \frac{\Delta h_1}{\Delta q_b} \right|_{h_2 cte} = \frac{K_{12}K_{21} - K_{11}K_{22}}{K_{21}} \quad (3.5)$$

$$K_{C21} = \left. \frac{\Delta h_2}{\Delta q_a} \right|_{h_1 cte} = \frac{K_{12}K_{21} - K_{11}K_{22}}{K_{12}} \quad (3.6)$$

$$K_{C22} = \left. \frac{\Delta h_2}{\Delta q_b} \right|_{h_1 cte} = \frac{K_{11}K_{22} - K_{12}K_{21}}{K_{11}} \quad (3.7)$$

$$K_C = \begin{pmatrix} K_{C11} & K_{C12} \\ K_{C21} & K_{C22} \end{pmatrix} \quad (3.8)$$

$$\Lambda = [\lambda_{ij}] = \left[\frac{K_{ij}}{K_{Cij}} \right] \quad (3.9)$$

Aplicado sobre el punto de trabajo donde $h_1=0.5955$, $h_2=0.6616$, $h_3=0.5384$, $h_4=0.7682$ $q_a=1.7602$, $q_b=1.8072$ llegamos a

$$K = \begin{pmatrix} 0.2216 & 0.4427 \\ 0.4733 & 0.2706 \end{pmatrix} \quad K_C = \begin{pmatrix} -0.5527 & 0.316 \\ 0.3378 & -0.6749 \end{pmatrix} \quad (3.10)$$

$$\Lambda = \begin{pmatrix} -0.4 & 1.4 \\ 1.4 & -0.4 \end{pmatrix}$$

Lo que implica en este caso h_1 emparejada con q_b y h_2 emparejada con q_a .

$$H_1 \leftrightarrow Q_b$$

$$H_2 \leftrightarrow Q_a$$

3.4 Cálculo de controlador proporcional (P)

Se aplica dos controladores proporcionales sobre el modelo no lineal en bucle cerrado.

Donde $C_1(s) = K_C$ $C_2(s) = K_C$

Como paso previo ajustamos la constante K_C sobre el modelo lineal del sistema. En este caso haremos que h_1 dependa de la entrada q_a , mientras que h_2 de q_b , aunque se trate de un emparejamiento incorrecto.

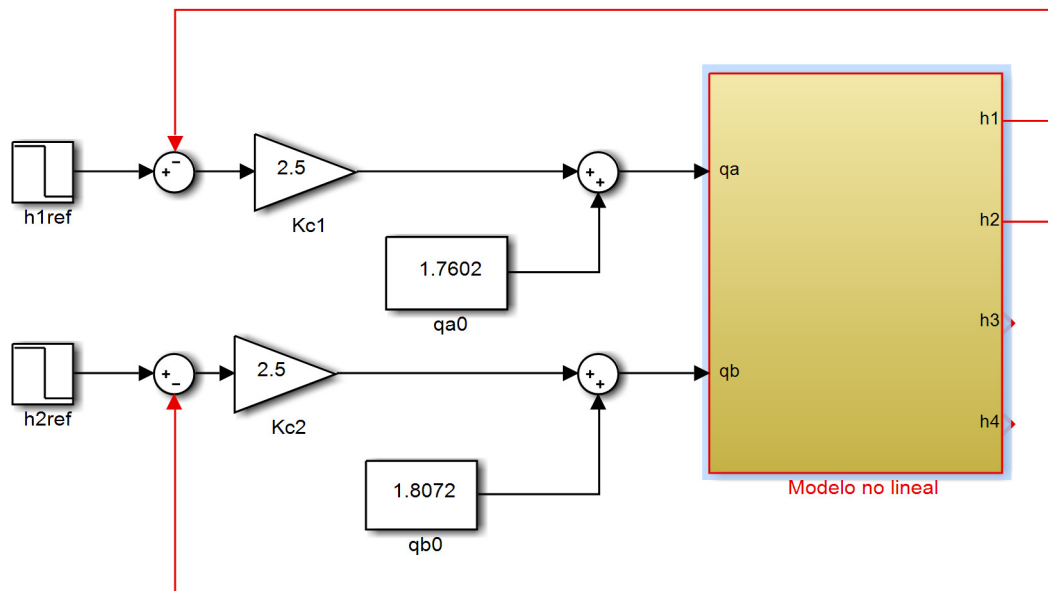


Fig 3.2. Esquema de simulink del controlador proporcional de la planta de los cuatro tanques realimentado

En la figura 3.3 se muestra que tiene error en régimen permanente nulo sólo cuando se resuelve el problema de seguimiento para el punto de funcionamiento diseñado. En este caso al ser diseñado para $h_1 = 0.5955$, $h_2 = 0.6616$, $h_3 = 0.5384$, $h_4 = 0.7682$

$q_a = 1.7602$, $q_b = 1.8072$ alcanza el punto objetivo sin error permanente en el segundo escalón, que corresponde con el punto de funcionamiento diseñado.

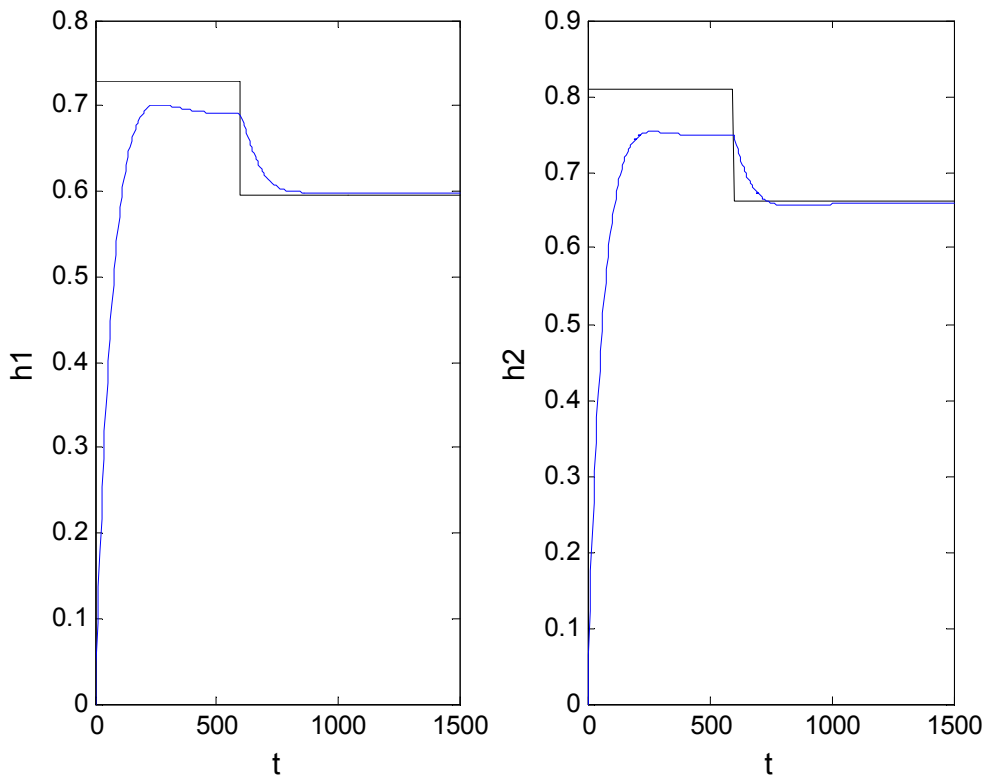


Fig 3.3. Representación de las salidas h_1 y h_2 de la planta de los cuatro tanques controlado por un controlador proporcional en bucle cerrado

3.6 Control proporcional integral (PI) por cancelación de dinámica

3.6.1 PI sin cambio de parámetros

Se decide calcular dos controladores proporcionales integrativos que controlen los valores de los actuadores q_a y q_b . De esta forma haciendo uso del Método de Bristol asignamos que h_1 dependa de la entrada q_b , mientras que h_2 de q_a .

$$C_1(s) = K_{C1} \frac{1 + T_{i1}s}{T_{i1}s} \quad C_2(s) = K_{C2} \frac{1 + T_{i2}s}{T_{i2}s} \quad (3.11)$$

Un método sencillo para calcular T_i es igualar $T_i = \tau$. Observando las funciones de transferencias obtenidas en el apartado anterior para el punto de trabajo $h_1 = 0.5955$, $h_2 = 0.6616$, $h_3 = 0.5384$, $h_4 = 0.7682$ $q_a = 1.7602$, $q_b = 1.8072$

$$G_H = \begin{pmatrix} \frac{0.2216}{1+79.2s} & \frac{0.4427}{1+199s} \\ \frac{0.4733}{1+218.5s} & \frac{0.2706}{1+72.65s} \\ \frac{0.596}{1+107s} & 0 \\ 0 & \frac{0.872}{1+202.5s} \end{pmatrix} \quad (3.12)$$

Calculamos $T_{i1}=199$ y $T_{i2}=218.5$ y K_C lo modificamos a partir de $\tau_{bc}=0.7\tau$

Siendo $G_{bc}(s) = \frac{1}{1 + \frac{\tau}{KK_C}s} \quad K_C = \frac{1}{0.7K} \quad (3.13)$

Llegamos a $K_{C1} = K_{C2} = 2.5$

Quedando de la forma $C_1(s) = 2.5 \frac{1+199s}{199s} \quad C_2(s) = 2.5 \frac{1+218.5s}{218.5s} \quad (3.14)$

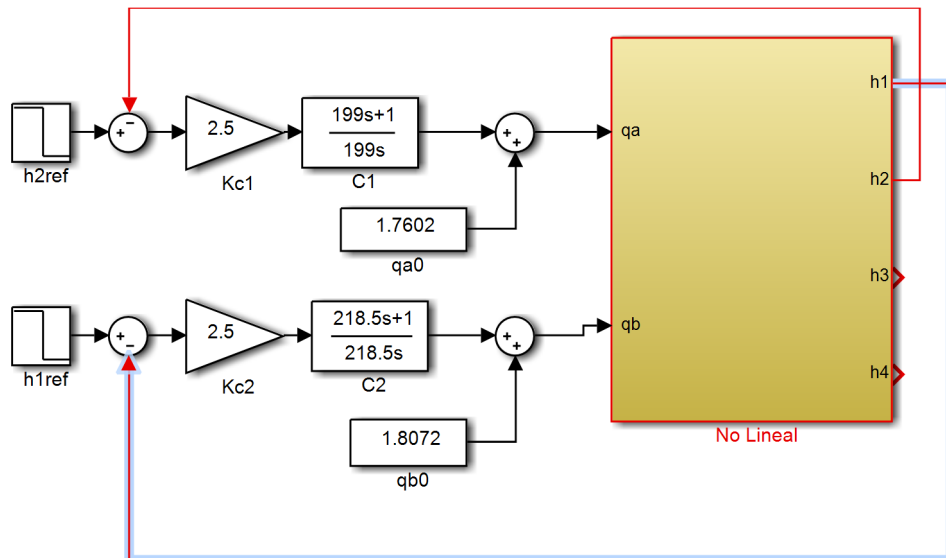


Fig 3.4. Modelo de simulink del control PI donde h_1 depende de q_b , mientras que h_2 de q_a .

En la figura 3.5 se quiere demostrar que el control PI tomando los emparejamientos $h_1 \rightarrow q_b$ y $h_2 \rightarrow q_a$ hace que nuestro sistema sea estable y alcance el punto objetivo, mientras que el control PI por medio del emparejamiento $h_1 \rightarrow q_a$ y $h_2 \rightarrow q_b$ hace que nuestro sistema se inestabilice. Hecho que se demuestra al observar la evolución de las salidas h_1 y h_2 con los diferentes emparejamientos.

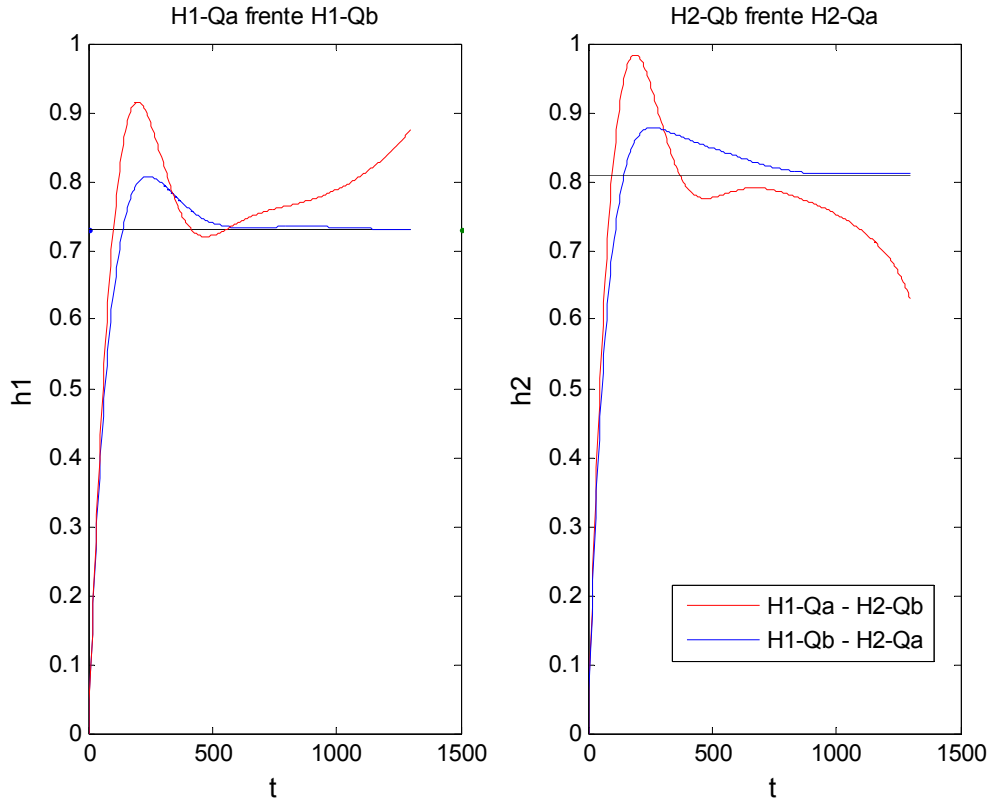


Fig 3.5. Comparativa de h_1 y h_2 usando o no los emparejamientos recomendados del método de Bristol

3.5.1 PI con cambio de parámetros

El problema que motiva este apartado es la necesidad de llevar a nuestro sistema a un punto de funcionamiento diferente para el que ha sido diseñado. Así pues, el objetivo consiste en probar el comportamiento de los controladores deseados frente a un cambio de referencia en las alturas y comparar la respuesta del sistema mediante dos métodos. Por una parte usando un conjunto de controladores diseñados que cambian sus parámetros K_C y T_i dependiendo la altura (puntos de trabajo) que se quiera alcanzar y por otra un único conjunto de controladores diseñado para un punto de trabajo.

Los dos puntos que queremos alcanzar son

a) $h_1 = 0.5955$, $h_2 = 0.6616$, $h_3 = 0.5384$, $h_4 = 0.7682$ $q_a = 1.7602$, $q_b = 1.8072$

b) $h_1 = 0.7293$, $h_2 = 0.8102$, $h_3 = 0.6594$, $h_4 = 0.9408$ $q_a = 1.904$, $q_b = 2$

Al igual que hacíamos en el apartado anterior para un punto de funcionamiento se procede en primer lugar calculando sus funciones de transferencia

$$G_H^1 = \begin{pmatrix} \frac{0.2216}{1+79.2s} & \frac{0.4427}{1+199s} \\ \frac{0.4733}{1+218.5s} & \frac{0.2706}{1+72.65s} \\ \frac{0.596}{1+107s} & 0 \\ 0 & \frac{0.872}{1+202.5s} \end{pmatrix} \quad G_H^2 = \begin{pmatrix} \frac{0.245}{1+86s} & \frac{0.49}{1+215s} \\ \frac{0.525}{1+243s} & \frac{0.3}{1+80s} \\ \frac{0.66}{1+118s} & 0 \\ 0 & \frac{0.965}{1+147s} \end{pmatrix} \quad (3.15)$$

Y por cancelación dinámica el conjunto de controladores PI

$$C_1^1(s) = 2.5 \frac{1+199s}{199s} \quad C_2^1(s) = 2.5 \frac{1+218.5s}{218.5s} \quad C_1^2(s) = 3 \frac{1+215s}{215s} \quad C_2^2(s) = 3 \frac{1+243s}{243s} \quad (3.16)$$

Se designa a C_1^1 y C_2^1 al conjunto de controladores que están diseñados para alcanzar h_{ref}^1 y C_1^2 y C_2^2 para h_{ref}^2 . En la siguiente figura 3.6 se muestra el experimento que se quiere simular. Partiendo de un estado inicial $h_0 \in \mathbb{R}^4$ siendo $h_0 = \{h_{01}, h_{02}, h_{03}, h_{04}\}$ y alcanzar $h_{ref} \in \mathbb{R}^4$ siendo $h_{ref} = \{h_{ref1}, h_{ref2}, h_{ref3}, h_{ref4}\}$. Por un lado se hace uso de un único conjunto de controladores C_1^1 y C_2^1 (azul) y por otro dos conjunto de controladores C_1^1 y C_2^2 , C_1^2 (azul) y C_2^2 (rojo) para alcanzar dos puntos de referencia.

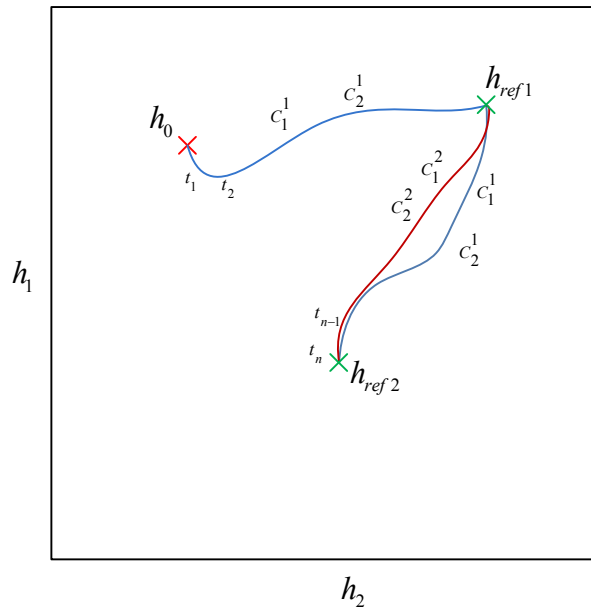


Fig 3.6. Problema de seguimiento (por simplicidad en la representación las figuras se representan en \mathbb{R}^2 con h_1 y h_2 , sin embargo se trata de \mathbb{R}^4 con h_1, h_2, h_3 y h_4)

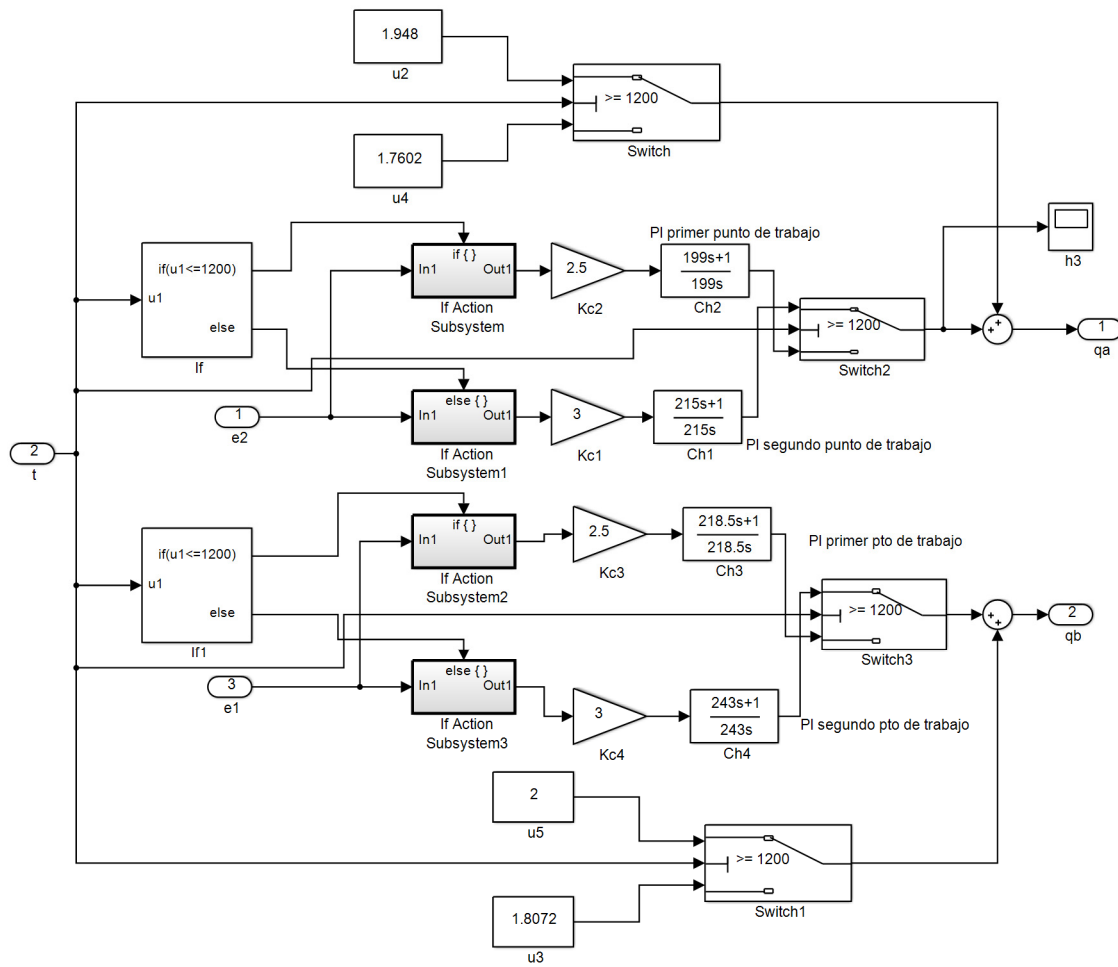


Fig 3.7. Esquema de simulink de la implementación de dos conjuntos de controladores PI, tal que C_1^1 (antes de 1200 seg.) y C_1^2 (a partir de 1200 seg.)

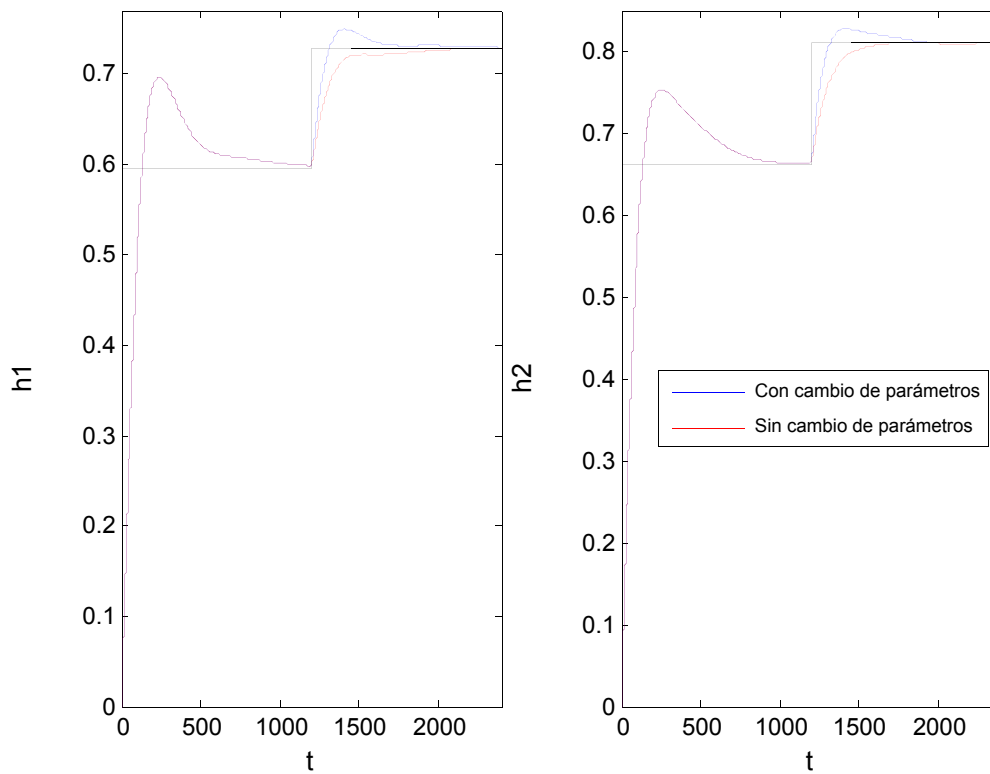


Fig 3.8. Comparativa de h_1 y h_2 usando G_{c1} y G_{c2} con y sin cambio de parámetros

A simple vista observando la figura 3.8 no podríamos decir que el conjunto de controladores C_1^2 no puede alcanzar h_{ref}^2 , de hecho no existen grandes diferencias entre las trayectorias de C_1^1 y C_1^2 . Por ello, en términos generales, no haría falta crear diferentes controladores PIs para alcanzar diferentes puntos objetivos, ya que la componente integral compensa los errores en régimen permanente y alcanza el punto objetivo.

3.6 Control anticipativo (feedforward)

3.6.1 Principios del control anticipativo [1, PP 181-184]

La idea básica del control anticipativo (también llamado control en adelanto o control por prealimentación (feedforward)) es la de medir las perturbaciones que inciden sobre la salida y actuar sobre el proceso inmediatamente, sin esperar a que dichas perturbaciones afecten a la salida (variable controlada). Para ello se utiliza un modelo del efecto de la perturbación sobre la salida para poder compensarlo.

Usando un modelo de la planta y un modelo de las perturbaciones que afectan a la salida, se tendría la tentación de pensar que sería posible un control perfecto sin necesidad de realimentar la variable controlada. Esto sería así porque el control

anticipativo no tendría que esperar a que hubiese error para empezar a corregirlo. En la práctica esto no funciona por las siguientes razones:

- No es posible medir todas las perturbaciones que afectan a la salida.
- Siempre existirá un error en la medida de las perturbaciones.
- El modelo del proceso y el de las perturbaciones siempre tendrá errores de modelado.
- A veces el controlador anticipativo resultante no es realizable. Esto sucede cuando el retraso entre la perturbación y la salida es inferior al que hay entre la salida y la entrada.

El esquema clásico que presenta este controlador es:

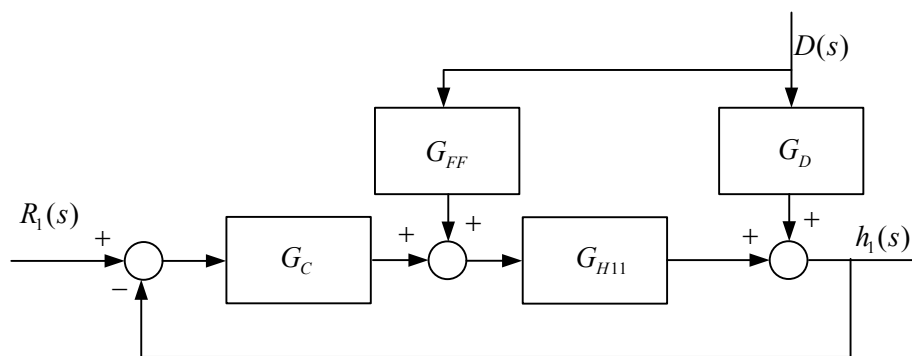


Fig 3.9. Sistema con perturbación a la salida controlado con un controlador anticipativo con control realimentado

3.6.2 Aplicación del control feedforward sobre la planta de los cuatro tanques

En este apartado se intenta abordar el problema del control de la planta de los cuatro tanques tomando h_1 que dependa de q_a y h_2 de q_b y el caudal tomar el caudal que desagua desde los depósitos superiores 3 y 4 en los inferiores 1 y 2 como perturbación. Con el fin de resolver este problema aplicamos los principios del controlador anticipativo (feedforward).

En el caso de la planta de los 4 tanques conocemos las perturbaciones, ya que existen sensores que miden tanto h_3 como h_4 y a partir de esos datos podemos obtener el caudal que desagua desde los depósitos superiores 3 y 4 en los inferiores 1 y 2. Así pues intentamos compensar los incrementos de las perturbaciones actuando de la siguiente forma:

$$\begin{aligned}
 \Delta h_3 \uparrow &\rightarrow \Delta q_a \downarrow & \Delta h_4 \uparrow &\rightarrow \Delta q_b \downarrow \\
 \Delta h_3 \downarrow &\rightarrow \Delta q_a \uparrow & \Delta h_4 \downarrow &\rightarrow \Delta q_b \uparrow
 \end{aligned}
 \quad (3.17)$$

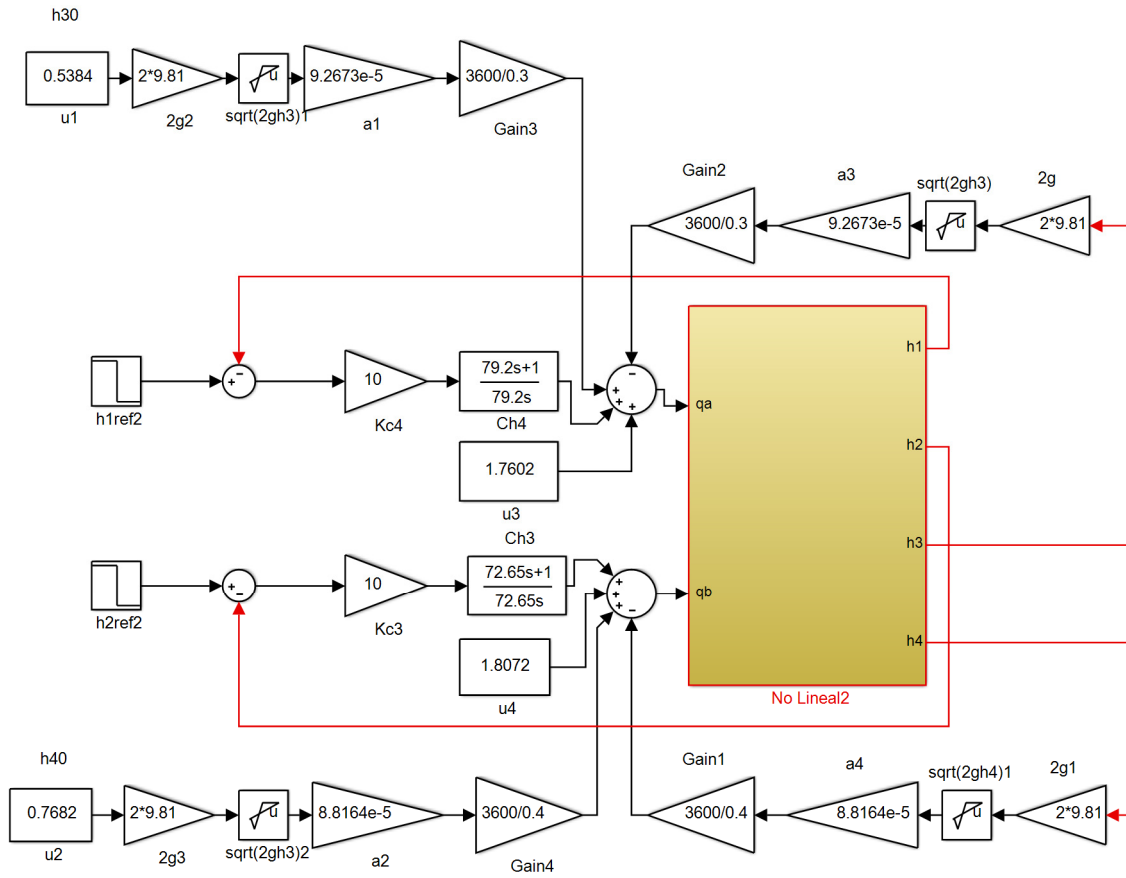


Fig 3.10. Esquema de simulink del control feedforward de la planta de los cuatro tanques realimentado

Sin embargo, esta compensación de las perturbaciones no consigue que se alcance el punto objetivo por lo que no parece suficiente para que pueda controlar este sistema, tal y como se muestra en la figura 3.21.

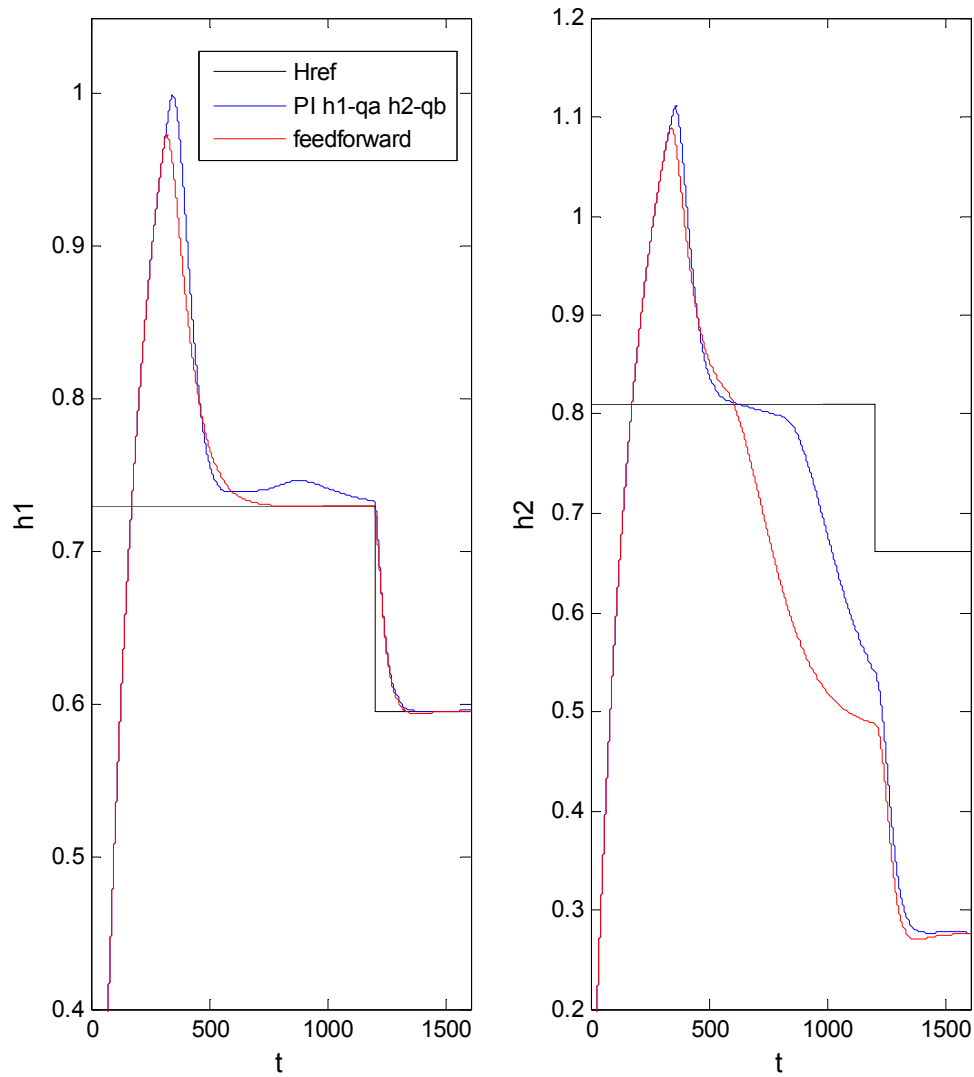


Fig 3.21. Comparativa de las salidas h_1 , h_2 de la planta de los cuatro tanques controlado por un controlador feedforward y un controlador integrativo en bucle cerrado

3.7 Regulador óptimo cuadrático LQR

3.7.1 Introducción al control óptimo LQR [1, PP 119,120]

En este apartado nos centraremos en el caso particular del control LQR con horizonte infinito, también conocido como LQR de régimen permanente.

Las estrategias de control óptimo calculan la ley de control de manera que se optimiza una cierta medida del rendimiento del controlador. Se parte de un sistema descrito por

$$x(k+1) = Gx(k) + Hu(k) \quad (3.18)$$

El objetivo es calcular una ley de control

$$u(k) = -Kx(k) \quad (3.19)$$

de tal manera que se minimiza el funcional (que expresa un índice de funcionamiento)

$$J = \frac{1}{2} \sum_{k=0}^{\infty} (x^*(k) Q x(k) + u^*(k) R u(k)) \quad (3.20)$$

siendo Q y R matrices de ponderación que cumplen que $Q^* = Q > 0$, $R^* = R > 0$

Nótese que este índice de funcionamiento pondera la diferencia entre el estado y el origen el instante inicial, hasta un tiempo infinito. Por tanto, cuanto más rápido se llegue al origen menor valor de J se tendrá. Esto implica que al minimizarse J , se encontrará la ley de control que lleva el estado al origen más rápidamente y manteniéndolo siempre lo más cerca posible del origen. Por otra parte, se observa que en el funcional hay otro término que pondera el valor de la secuencia de señales de actuación. Este término impide que se obtenga una ley de control que lleve el estado al origen a expensas de una actuación muy grande. Al minimizarse J , por tanto, se conseguirá una ley de control que por una parte acerque el estado al origen lo más rápido posible, pero manteniendo un nivel de actuaciones moderado, encontrándose por tanto, una solución de compromiso entre el rendimiento del controlador y su nivel de actuación. El sentido de este compromiso puede venir dictado por diferentes razones, como por ejemplo moderar el gasto de energía o combustible necesario para proporcionar la señal de actuación. Existen razones más sutiles pero no por ello menos importantes para incorporar esta ponderación del esfuerzo de control. Por ejemplo, cuando existen discrepancias entre el modelo del sistema y su dinámica real (algo que ocurre casi siempre, pues los modelos matemáticos no pueden recoger todas las complejidades de los sistemas o procesos reales) esta ponderación del esfuerzo de control resulta en un sistema más estable.

Para calcular la ley de control que minimiza el índice se define una matriz P que satisface la siguiente ecuación de Riccati:

$$P = Q + G^* P G - G^* P H (R + H^* P H)^{-1} H^* P G \quad (3.21)$$

La solución de esta ecuación es una matriz P que es hermética y definida positiva. Se demuestra que la matriz

$$K = (R + H^* P H)^{-1} H^* P G \quad (3.22)$$

es la que minimiza el índice J mediante la ley de control

$$u(k) = -(R + H^* P H)^{-1} H^* P G x(k) \quad (3.23)$$

La ecuación de estado del sistema en bucle cerrado será por tanto:

$$x(k+1) = (I + H R^{-1} H^* P)^{-1} G x(k) \quad (3.24)$$

3.7.2 Aplicación del control LQR a la planta de los cuatro tanques [OGATA,98]

Partiendo del modelo en el espacio de estado

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (3.25)$$

Donde

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} \dot{h}_1 \\ \dot{h}_2 \\ \dot{h}_3 \\ \dot{h}_4 \end{pmatrix} \quad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} \quad \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} q_a \\ q_b \end{pmatrix} \quad (3.26)$$

Llegamos a

$$\begin{pmatrix} \dot{h}_1 \\ \dot{h}_2 \\ \dot{h}_3 \\ \dot{h}_4 \end{pmatrix} = \frac{1}{A^*} \begin{pmatrix} -A_1 & 0 & A_3 & 0 \\ 0 & -A_2 & 0 & A_4 \\ 0 & 0 & -A_3 & 0 \\ 0 & 0 & 0 & -A_4 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} + \frac{1}{3600 A^*} \begin{pmatrix} \gamma_a & 0 \\ 0 & \gamma_b \\ 0 & 1 - \gamma_b \\ 1 - \gamma_b & 0 \end{pmatrix} \begin{pmatrix} q_a \\ q_b \end{pmatrix} \quad (3.27)$$

Siendo la sección de todos los tanques

$$A_i = a_i \sqrt{2gh_i} \quad \forall i = 1, 2, 3, 4 \quad \text{y} \quad A^* = 0.03$$

Siendo las componentes del espacio de estado

$$A = \frac{1}{A^*} \begin{pmatrix} -A_1 & 0 & A_3 & 0 \\ 0 & -A_2 & 0 & A_4 \\ 0 & 0 & -A_3 & 0 \\ 0 & 0 & 0 & -A_4 \end{pmatrix} \quad B = \frac{1}{3600 A^*} \begin{pmatrix} \gamma_a & 0 \\ 0 & \gamma_b \\ 0 & 1 - \gamma_b \\ 1 - \gamma_b & 0 \end{pmatrix} \quad (3.28)$$

$$Q = \begin{pmatrix} 4000 & 0 & 0 & 0 \\ 0 & 4000 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{pmatrix} \quad R = \begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \quad (3.29)$$

Usamos la función de *lqr* de la librería de Matlab para resolver este problema

$$K1 = lqr(A, B, Q, R); \quad (3.30)$$

En la implementación que se ha realizado sobre la planta de los 4 tanques podemos observar que se ha obligado a que se alcance de manera más rápida los valores de referencia de las alturas de los depósitos en detrimento del valor de los caudales al ponderar con valores más altos las componentes de Q que R .

El controlador para el punto de funcionamiento

$$h_1 = 0.5955, h_2 = 0.6616, h_3 = 0.5384, h_4 = 0.7682 \quad q_a = 1.7602, q_b = 1.8072$$

$$K_C = \begin{pmatrix} 11.6042 & 6.6845 & -4.3096 & 3.5267 \\ 7.1535 & 13.6119 & 4.1264 & -2.9397 \end{pmatrix} \quad (3.31)$$

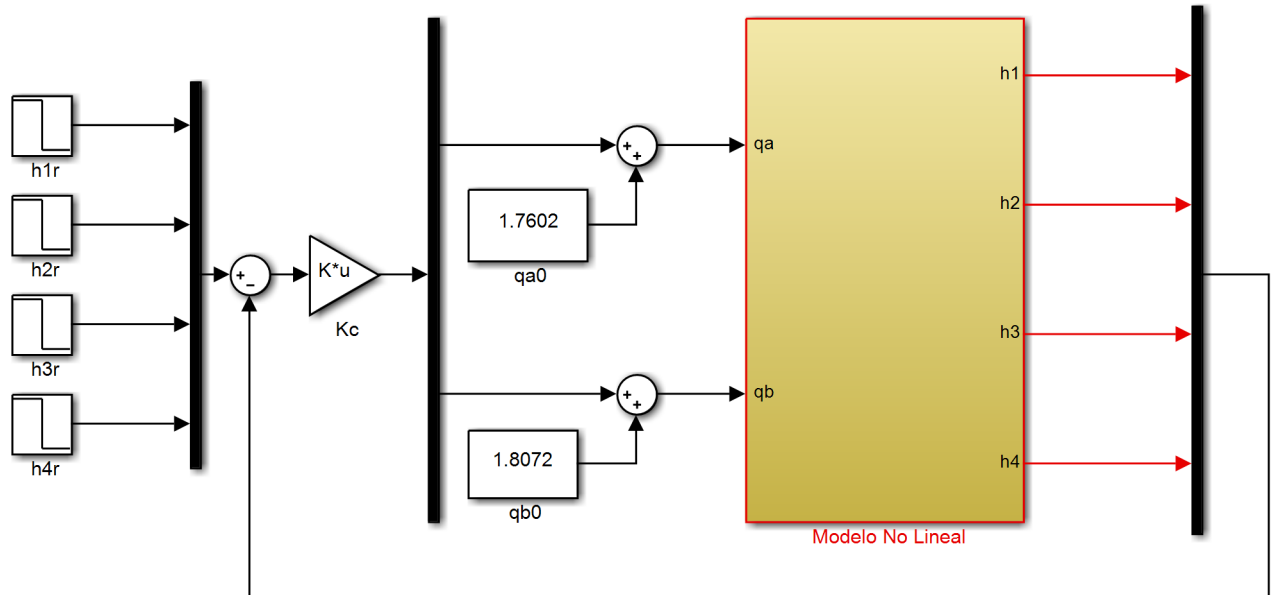


Fig 3.22. Esquema de simulink del esquema de control del controlador LQR en bucle cerrado

Hay que señalar que mientras en el controlador proporcional o el proporcional integrativo no intervienen el error de h_3 y h_4 , ya que sólo nos servíamos del $e_1 = h_{ref1} - h_1$ y $e_2 = h_{ref2} - h_2$, en el controlador LQR q_a y q_b dependen de h_1 , h_2 , h_3 y h_4 .

En la figura 3.23 se muestra el comportamiento del controlador LQR ante un problema de seguimiento para alcanzar dos puntos objetivos. El primero se trata del punto de funcionamiento para el que ha sido implementado y el segundo un punto de equilibrio cercano. Se observa que para el primer caso no tiene error en régimen permanente, mientras que en el segundo por el contrario sí. Presenta este error debido a que dicho controlador no tiene un factor integral que acumule los errores y de esta forma corregirlos.

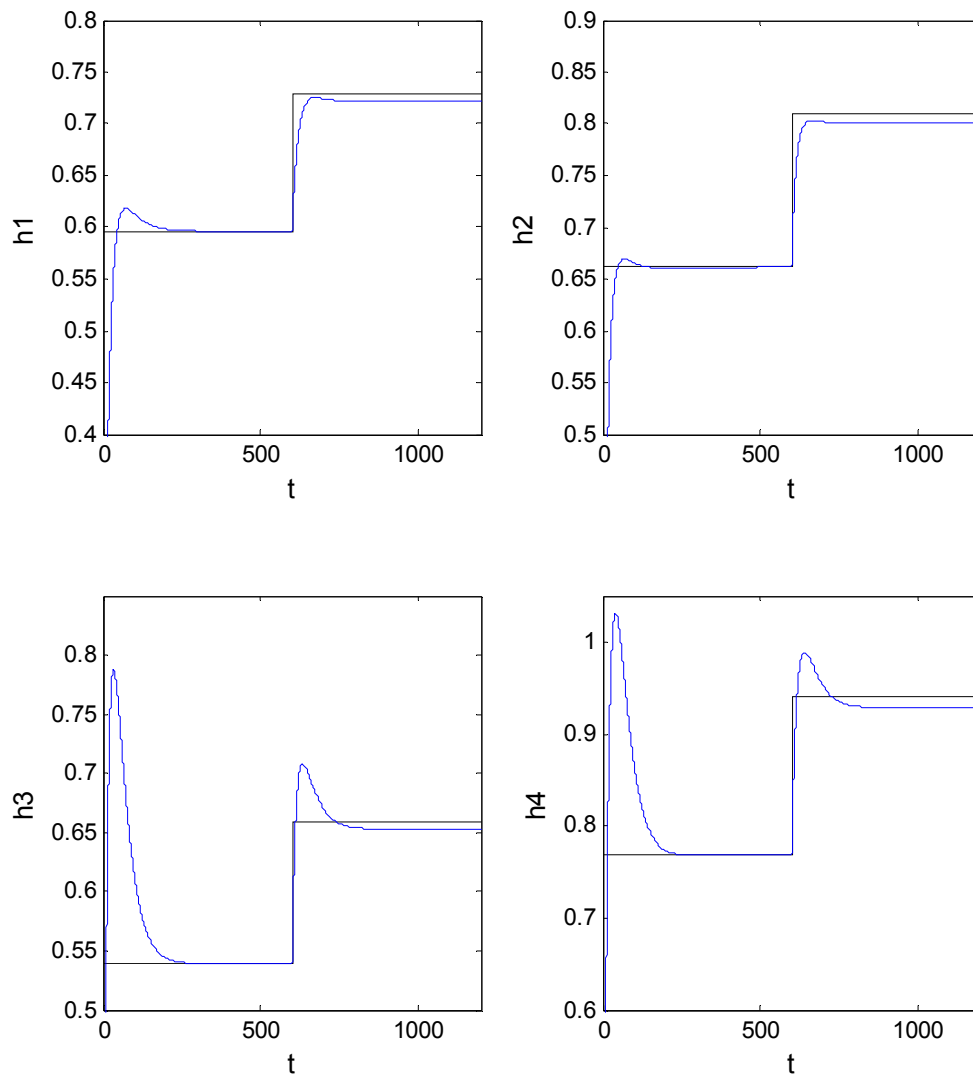


Fig 3.23. Representación de las salidas h_1 , h_2 , h_3 y h_4 de la planta de los cuatro tanques controlado por un controlador LQR en bucle cerrado

4. Base de datos

4.1 Composición de la base de datos

Una base de datos es un banco de información perteneciente a un mismo contexto y almacenados sistemáticamente para su posterior uso. Según la variabilidad de la base de datos se puede decir que se trata de estática o dinámica. Mientras que las base de datos estáticas son utilizadas primordialmente para la lectura de los datos que se han almacenado, en las dinámicas la información se modifica con el tiempo.

En nuestro caso se trata de una base de datos estática donde se almacena un histórico de las trayectorias llevadas a cabo por la planta de los 4 tanques en bucle cerrado con diferentes controladores durante un periodo de tiempo t . Estas trayectorias son generadas haciendo uso del modelo no lineal del sistema. No obstante, en la práctica se tratarían de datos que se han ido almacenando a lo largo de los años en la planta real. En un caso ideal donde el modelo sea equivalente a la planta real se extraerían los mismos resultados en el caso que partamos de un mismo estado inicial. Sin embargo, esto no ocurre en la realidad y siempre existe siempre un error asociado a la diferencia entre la planta y el sistema no lineal. Por tanto se intentan simular las diferentes situaciones de la planta haciendo diferentes simulaciones de diferentes estados iniciales.

Es importante señalar que se ha implementado un DbPC para regular un único punto objetivo. Así pues, existe una base de datos diferente para cada punto objetivo por lo que h_{ref1} y h_{ref2} es igual para todos los registros de una misma base de datos. En una ampliación del trabajo se trabajaría con diferentes h_{ref} para regular diferentes puntos.

En nuestros proyecto las trayectorias se generan a partir del control en bucle cerrado del modelo de los 4 tranques por medio de diferentes controladores. Como ya hemos explicado en el apartado anterior se trata de un controlador P, PI y LQR. Cada trayectoria parte de un estado inicial diferente $h_i^0 \in \mathbb{R}^4$ siendo $h_i^0 = \{h_{10}, h_{20}, h_{30}, h_{40}\}$ y convergen a $h_{ref} \in \mathbb{R}^4$ siendo $h_{ref} = \{h_{ref1}, h_{ref2}, h_{ref3}, h_{ref4}\}$. Se ha optado por establecer el tiempo de simulación de cada trayectoria en base a que sea suficientemente grande como para poder alcanzar la referencia deseada desde cualquier estado inicial. Dicho parámetro se ha establecido en $t = 2500$.

La base de datos que hemos generado almacena diferentes trayectorias posibles para alcanzar una altura determinada $h_{ref} \in \mathbb{R}^4$. Se trata de una matriz compuesta por 11 columnas y 1503000 filas. Cada trayectoria está compuesta por 500 registros donde se almacenan las salidas y entradas, alturas y acción de control aplicada, del estado actual de la trayectoria en un instante de tiempo. La base de datos no está almacenada de forma aleatoria, sino que todas las muestras de cada experimento se encuentran almacenadas de

forma consecutiva. Además cada registro está etiquetado de forma que se conozca a que experimento pertenece.

Tabla 4.1. Composición de la base de datos

Id	C	h_{ref1}	h_{ref2}	h_1	h_2	h_3	h_4	q_a	q_b	t
1	1	0.5955	0.6616	0.7067	0.8109	0.6167	0.9788	0.6213	1.2634	0
1	1	0.5955	0.6616	0.7067	0.8109	0.6167	0.9788	0.6213	1.2634	5
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
1	1	0.5955	0.6616	0.5956	0.6614	0.5382	0.7684	1.7601	1.8073	2500
2	1	0.5955	0.6616	0.9896	0.4564	0.7892	1.0231	0.3245	0.9848	0
2	1	0.5955	0.6616	0.9896	0.4564	0.7892	1.0231	0.3245	0.9848	5
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
2	1	0.5955	0.6616	0.5956	0.6614	0.5382	0.7684	1.7601	1.8073	2500
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
1000	1	0.5955	0.6616	0.5954	0.6616	0.5385	0.7684	1.7602	1.8071	2500
1	2	0.5955	0.6616	0.7067	0.8109	0.6167	0.9788	0.6213	1.2634	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
1000	3	0.5955	0.6616	0.5955	0.6616	0.5384	0.7682	1.7602	1.8072	2500

Como se muestra en la tabla 4.1 en la primera columna se observa el registro de los experimentos realizados, en la segunda el tipo de controlador que se usa $P=1$, $PI=2$ y $LQR=3$, tercera y cuarta las referencias h_{ref1} y h_{ref2} , de la quinta a la décima las salidas h_1, h_2, h_3, h_4 y entradas q_a, q_b del sistema. Por último, el tiempo de simulación donde establecemos un tiempo de muestre de $T_m = 5$. No es necesario guardar en la base de datos ni h_{ref3} , h_{ref4} , q_a^* y q_b^* porque se pueden obtener del modelo dinámico forzando que sea un punto de equilibrio.

4.2 Métodos de cálculo de los estados iniciales

Un punto importante es la elección de los estados iniciales $h_i \in \mathbb{R}^4$ siendo $h_i = \{h_1, h_2, h_3, h_4\}$. Al ser casi imposible y poco eficiente almacenar las infinitas trayectorias que convergen a h_{ref} se decide crear esas trayectorias partiendo de un número suficiente de estados iniciales. Este número suficiente de estados iniciales se discutirá en la sección 6 donde se analizará el comportamiento y bondad de diferentes controladores predictivos que toman diferentes bases de datos con diferente número de estados iniciales. Así pues, estos estados iniciales se han elegido mediante dos algoritmos.

Ambos algoritmos tienen en común que calculan estados iniciales de puntos de equilibrio, en otras palabras, se fuerza a que $h_i^0 = \{h_{10}, h_{20}, h_{30}, h_{40}\}$ cumpla la siguiente ecuación haciendo las derivadas nulas.

$$\frac{dh_1}{dt} = \frac{dh_2}{dt} = \frac{dh_3}{dt} = \frac{dh_4}{dt} = 0 \quad (4.1)$$

$$\begin{aligned} A \frac{dh_1}{dt} &= -a_1 \sqrt{2gh_1} + a_3 \sqrt{2gh_3} + \gamma_a \frac{q_a}{3600} \\ A \frac{dh_2}{dt} &= -a_2 \sqrt{2gh_2} + a_4 \sqrt{2gh_4} + \gamma_b \frac{q_b}{3600} \\ A \frac{dh_3}{dt} &= -a_3 \sqrt{2gh_3} + (1 - \gamma_b) \frac{q_b}{3600} \\ A \frac{dh_4}{dt} &= -a_4 \sqrt{2gh_4} + (1 - \gamma_a) \frac{q_a}{3600} \end{aligned} \quad (4.2)$$

De esta forma los únicos parámetros son h_{10}, h_{20} que calculan h_{30}, h_{40} por medio de la resolución de las ecuaciones no lineales en un punto de equilibrio.

a) Método Grid

Éste consiste en tomar h_{10}, h_{20} de forma distribuida valores entre $h = [0, 1.4]$. Para obtener todas las combinaciones posibles hacemos dos bucles desde 0 hasta 1.4 con un paso de τ dependiendo del número de puntos iniciales que queramos que tenga nuestra base de datos.

b) Método Rand

Se trata de obtener h_{10}, h_{20} por medio de la función *rand* de las librerías de Matlab. Para aplicarlo a nuestro problema sería $h_{k0} = 1.4 \cdot \text{rand}(1) \quad \forall \quad k = 1, 2$.

Es importante resaltar que si en cualquiera de los casos no imponemos que $h_{30} \leq 1.4$ y $h_{40} \leq 1.4$ la densidad de estados iniciales de nuestro espacio de trabajo $[0, 1.4]^m$ se ve alterada. Mientras que en la figura 2.1 no se han impuesto restricciones, donde se observa claramente una densidad de estados iniciales menor, en el caso de la figura 2.2 si se han impuesto las restricciones. De esta forma al restringir que $h_{30} \leq 1.4$ y $h_{40} \leq 1.4$ todos los estados iniciales se encuentran dentro del espacio de trabajo. Este cambio provoca que nuestro controlador predictivo tenga mayor número de estados iniciales candidatos $h_i \in \mathbb{R}^4$ siendo $h_i = \{h_1, h_2, h_3, h_4\}$ próximo al estado actual y por tanto su comportamiento (hecho que se explicará en profundidad en la siguiente sección).

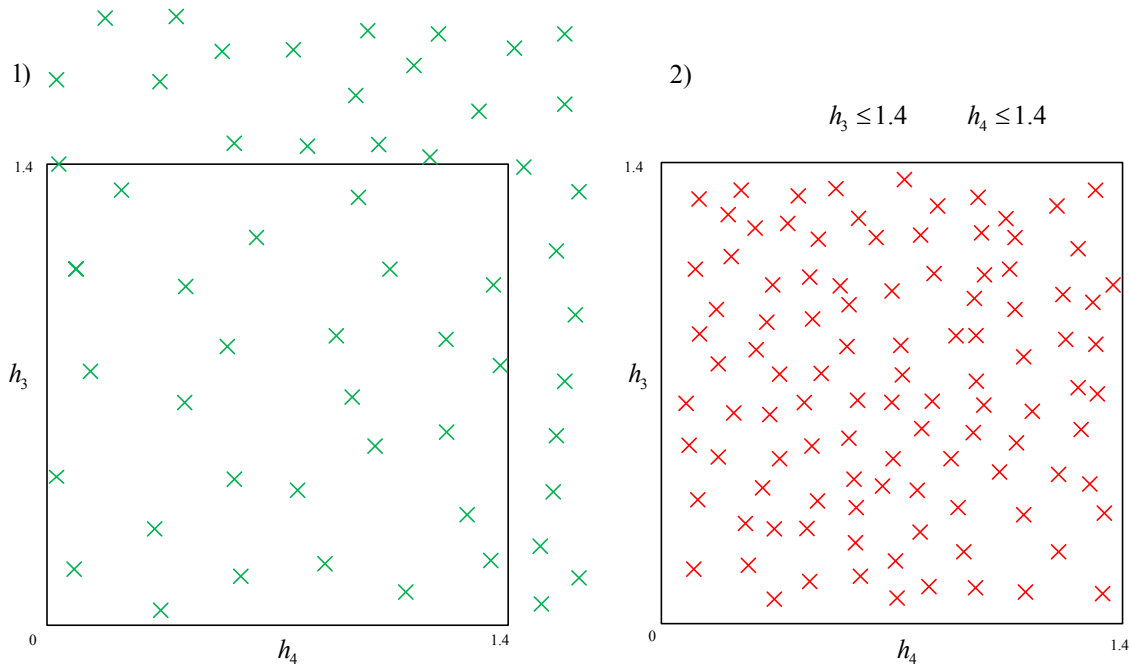


Fig 4.1. Estados iniciales en la base de datos

Una vez tenemos los puntos iniciales de nuestras trayectorias pasamos a resolver nuestro problema de regulación con tres controladores diferentes: P (Proporcional), PI (Proporcional Integrativo) y LQR (Linear-quadratic regulator). Estos se ejecutan en un mismo archivo de simulink *proyecto03.mdl* usando la función *sim*.

Una vez introducidos los parámetros de h_{ref1} , h_{ref2} y Ni (número de estados iniciales) el algoritmo pasa a calcular las diferentes simulaciones. En primer lugar, calcula las Ni trayectorias desde los Ni estados iniciales hasta h_{ref} con el controlador P. Luego, procede de la misma forma, pero esta vez usando el controlador PI y LQR. Y por último, la matriz que se ha generado la guardamos con la función *save* para poder cagarlas cuando hagamos uso del controlador predictivo.

Como particularidad del esquema de simulink que se muestra en la siguiente figura 4.2 se usa un bloque llamado *Multiport Switch* que permite que actúe un controlador u otro dependiendo de la constante *entrada*.

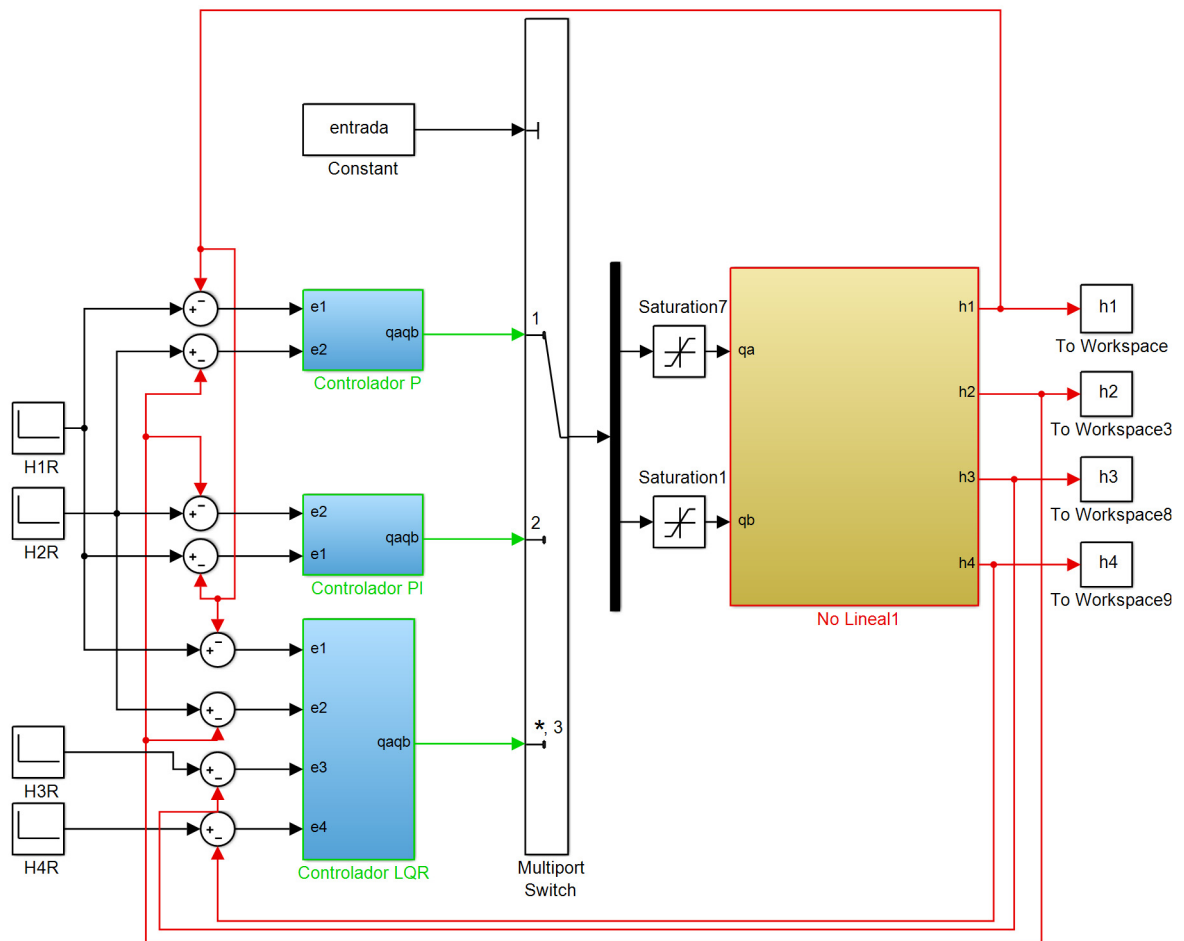


Fig 4.2. Modelo de simulink utilizado para la simulación de las diferentes trayectorias que se almacenan en la base de datos

5 Control predictivo basado en base de datos

5.1 Estado del arte

Antes de pasar a explicar cómo ha sido desarrollado el control predictivo mediante base de datos se tratará de hacer un repaso de las técnicas relacionadas con la aplicada en este proyecto, así como las diferencias más notables.

5.1.1 Perspectiva histórica [1, PP 235-236]

El control predictivo basado en modelo (MPC) se desarrolló en base a dos líneas básicas. Por un lado, a finales de los años setenta surgieron diversos algoritmos que usaban explícitamente un modelo dinámico del proceso para predecir el efecto de las acciones de control futuras en la salida, las cuales eran determinadas minimizando el error predicho sujeto a restricciones de operación. La optimización se repetía en cada instante de muestreo con información actualizada del proceso. Estas formulaciones eran de naturaleza heurística y algorítmica e intentaban aprovechar el creciente potencial de los computadores digitales por aquella época.

Rápidamente el MPC adquirió gran popularidad en las industrias de procesos químicos principalmente debido a la simplicidad del algoritmo y al uso del modelo de respuesta impulsional o en escalón, que aunque posea muchos más parámetros que las formulaciones en el espacio de estados o función de transferencia suele ser preferido por ser intuitivo y necesitar menos información a priori para identificar. La mayoría de las aplicaciones fueron llevadas a cabo sobre sistemas multivariables incluyendo restricciones. Los algoritmos utilizados fueron principalmente el IDCOM (identification-command) y el DMC (control con matriz dinámica, dynamic matrix control).

Independientemente fue surgiendo otra línea de trabajo en torno a las ideas del control adaptativo, desarrollando estrategias esencialmente para procesos monovariables formuladas con modelos entrada/salida. En este contexto se extendieron las ideas del controlador de mínima varianza y se desarrolló el control predictivo generalizado, generalized predictive control, (GPC) que es uno de los métodos más populares en la actualidad.

5.1.2 Conceptos básicos de control predictivo (MPC) [1, PP 236-237]

El control predictivo basado en modelo es un conjunto de técnicas que usan un modelo para predecir la salida del proceso a lo largo de un intervalo de tiempo de muestreo futuros (horizonte temporal). Para ello calcula la mejor secuencia futura de señales de control minimizando una función de coste.

Las ideas que aparecen en mayor o menor medida en toda la familia de controladores predictivos son básicamente:

- Uso explícito de un modelo para predecir la salida del proceso en futuros instantes de tiempo (horizonte).
- Cálculo de las señales de control minimizando una cierta función objetivo.
- Estrategia deslizante, de forma que en cada instante el horizonte se va desplazando hacia el futuro, lo que implica aplicar la primera señal de control en cada instante y desechar el resto, repitiendo el cálculo en cada instante de muestreo.

Los distintos algoritmos de MPC difieren entre sí casi exclusivamente en el modelo usado para representar el proceso y los ruidos y en la función de coste a minimizar. Aunque las diferencias puedan parecer pequeñas a priori, pueden provocar distintos comportamientos en bucle cerrado, siendo críticas para el éxito de un determinado algoritmo en una determinada aplicación.

- El MPC presenta una serie de ventajas sobre otros métodos, entre las que destacan:
- Resulta particularmente atractivo para personal sin un conocimiento profundo de control, puesto que los conceptos resultan muy intuitivos, a la vez que la sintonización es relativamente fácil.
- Puede ser usado para controlar una gran variedad de procesos, desde aquellos con dinámica relativamente simple hasta otros más complejos incluyendo sistemas con grandes retardos, de fase no mínima o inestables.
- Permite tratar con facilidad el caso multivariable.
- Posee intrínsecamente compensación del retardo.
- Resulta conceptualmente simple la extensión al tratamiento de restricciones, que pueden ser incluidas de forma sistemática durante el proceso de diseño.
- Es muy útil cuando se conocen las futuras referencias (robótica o procesos en
 - batch).
- Es una metodología completamente abierta basada en algunos principios básicos que permite futuras extensiones.

Pero, lógicamente, también presenta inconvenientes. Uno de ellos es la carga de cálculo necesaria para la resolución de algunos algoritmos. Pero quizás el mayor inconveniente venga marcado por la necesidad de disponer de un modelo apropiado del proceso. El algoritmo de diseño está basado en el conocimiento previo del modelo y es independiente de éste, pero resulta evidente que las prestaciones obtenidas dependerán de las discrepancias existentes entre el proceso real y el modelo usado.

5.1.3 Diferencias básicas del MPC y DbPC

Tanto el MPC como el DbPC comparten una misma estrategia de control (fig 5.1), esto es, ambos controladores predictivos toman como entrada las referencias y las medidas de los sensores del sistema a controlar y genera un salida que son los valores de los actuadores del sistema.

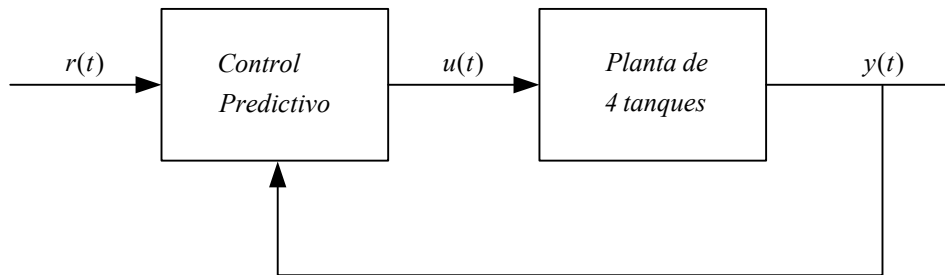


Fig 5.1. Esquema de control de controlador predictivo

Asimismo existen ideas que se presentan que la familia de controladores predictivos que comparten tanto el MPC como el DbPC. Éstas son el cálculo de las señales de control minimizando una cierta función objetivo y el uso de una estrategia deslizante, de forma que en cada instante el horizonte se va desplazando hacia el futuro, repitiendo el cálculo en cada instante de muestreo.

En la figura 5.2 se muestran los dos esquemas de control de ambos controladores. La diferencia más notable es que el controlador MPC hace uso del modelo del sistema, mientras que el DbPC de una base de datos. Del mismo modo el MPC utiliza el modelo para predecir la salida del proceso a lo largo de un intervalo de tiempo de muestreo futuros y toma normalmente sólo la primera componente de los controladores futuros como valor en el actuador del sistema. Mientras tanto el DbPC almacena en su base de datos las trayectorias de las cuales toma una solución óptima dependiendo del estado actual y de la referencia que queramos alcanzar. El problema del DbPC es qué conjunto de trayectorias se ha de tomar bajo estudio para obtener un valor de actuadores óptimos.

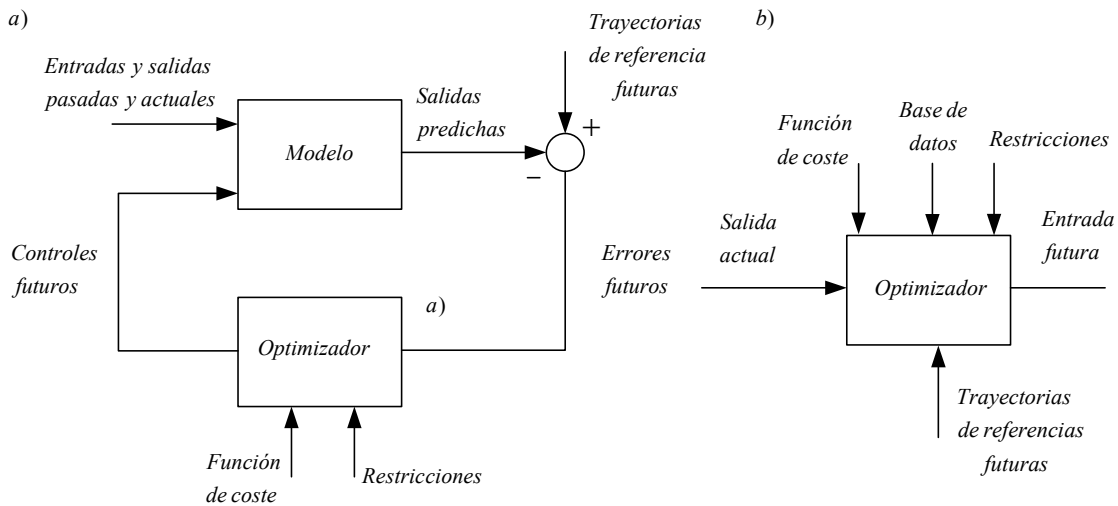


Fig 5.2. a) Esquema de MPC b) Esquema de control predictivo basado en datos

5.2 Principio del controlador predictivo basado en datos

El control predictivo mediante base de datos es una estrategia de control que nos permite resolver el problema de seguimiento a partir de los datos proporcionados por trayectorias almacenadas en una base de datos. Como hemos explicado en el apartado X (bloque de la base de datos) esta base de datos guarda los históricos del comportamiento en bucle cerrado de varios controladores que resuelven el problema de seguimiento.

En primer lugar, se parte de un estado inicial $h_0 \in \mathbb{R}^4$, siendo $h_0 = \{h_{01}, h_{02}, h_{03}, h_{04}\}$, y queremos llevar el sistema al punto de operación que hemos definido como $h_{ref} \in \mathbb{R}^4$ siendo $h_{ref} = \{h_{ref1}, h_{ref2}, h_{ref3}, h_{ref4}\}$ y $q_a^*, q_b^* \in \mathbb{R}$. Luego, el estado actual, que parte del estado inicial, se actualiza cada instante de muestreo con el objetivo de obtener los valores óptimos de los actuadores y alcanzar referencias deseadas en nuestro problema de seguimiento.

El algoritmo sobre el que se ha implementado el controlador busca trayectorias de una duración de T periodos (horizonte temporal) a partir de un estado inicial $h_i \in \mathbb{R}^4$ siendo $h_i = \{h_1, h_2, h_3, h_4\}$ próximo al estado actual. Es importante señalar que este conjunto de estados iniciales próximos al estado actual pertenecen a las trayectorias almacenadas en la base de datos. A partir de ahora llamaremos trayectorias candidatas “i” a aquellas que parten de estado inicial $h_i \forall i = 1, \dots, N$ siendo N el número total de trayectorias candidatas bajo estudio. También es importante señalar que se restringe que entre las h_i no haya dos estados iniciales que pertenezcan a la misma trayectoria.

Para ilustrar dicho proceso se muestra la siguiente figura 5.3 que para alcanzar $h_{ref} \in \mathbb{R}^4$ desde el punto $h_0 \in \mathbb{R}^4$ es necesario el cálculo en cada instante de tiempo t de los valores de los actuadores q_a y q_b . Asimismo se representa los estados actuales de $h \forall t_1, \dots, t_n$ y el estado inicial h_0 de la trayectoria de nuestro controlador predictivo.

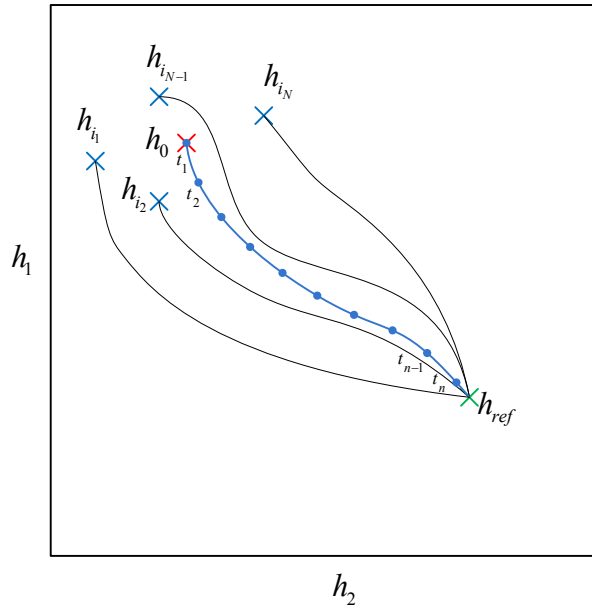


Fig 5.3. Problema de seguimiento (por simplicidad en la representación las figuras se representan en \mathbb{R}^2 con h_1 y h_2 , sin embargo se trata de \mathbb{R}^4 con h_1, h_2, h_3 y h_4)

Una vez obtenidas dichas trayectorias que distan como máximo ε al estado actual se calcula el índice de desempeño asociado a cada trayectoria candidata “i”.

El índice de desempeño se calcula como:

$$ID(t) = \int_0^T (J_i(h(\tau), q(\tau), c(\tau), p(\tau)) - J^*(c(\tau), p(\tau))) d\tau \quad (5.1)$$

En la que τ va desde $\tau = 0$ hasta $\tau = T$ siendo T el horizonte de predicción y la función J_i es el coste de etapa para el caso del problema de los 4 tanques, es decir, el coste de operación en un periodo de muestre definido como:

$$J_i(h, q, c, p) = q_{ia}^2 + c \cdot q_{ib}^2 + p \frac{V_{min}}{A(h_{i1} + h_{i2})} \quad \forall i = 1, \dots, N \quad (5.2)$$

Y siendo J^* el coste correspondiente al punto objetivo q_{ref} .

Esta función que mide la bondad de la trayectoria y nos sirve como base para calcular un nuevo controlador. Hay que insistir en la idea de que este índice de desempeño se calcula en línea y no se encuentra almacenado en la base de datos. De hecho T es un parámetro que podemos cambiar.

A continuación haciendo uso del valor de los índices de desempeño y distancia que existe entre el estado actual del sistema y el estado inicial de cada trayectoria decidimos la acción actual. Por último, aplica la acción actual de los actuadores sobre el sistema y actualiza tanto el valor del estado actual como el de los actuadores en cada periodo de muestreo t .

Para mostrar dicho algoritmo se presenta la figura 5.4 donde se muestre el esquema de control predictivo basado en base de datos. Como particularidad hay que señalar que en esa figura se muestra el caso donde se aplica programación lineal en el controlador para la búsqueda de una solución factible a nuestro problema, representado en la figura por el bloque de “Solución óptima: Minimiza el índice de desempeño”. No obstante, es posible que no sea una solución factible y se tenga que aumentar el número de trayectorias cercanas candidatas al estado actual. Una vez se obtenga una solución factible se calculará la acción futura.

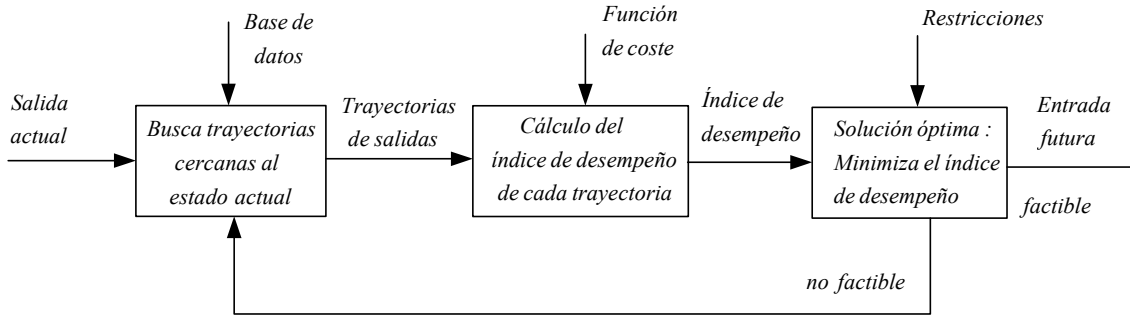


Fig 5.4. Esquema de control predictivo basado en base de datos

5.3. Algoritmos de resolución

En los próximos apartados se presentarán tres métodos diferentes que se han llevado a cabo en la programación del controlador predictivo basado en base de datos. Estos son:

- Método de mínimo índice de desempeño
- Método de índices ponderados
- Método mediante uso de programación lineal

5.3.1 Método de mínimo índice de desempeño

El método de mínimo índice de desempeño (ID), como su propio nombre indica, consiste en tomar el valor de los actuadores q_a y q_b de aquel punto que haga $\min J_i$ y si $d_i \leq \varepsilon$, siendo $d_i = \|h_i - h_0\|$ y siendo $h_i, h_0 \in \mathbb{R}^4$. h_i (estado inicial). Se escoge la trayectoria i con un coste mínimo de entre las trayectorias que distan como máximo ε al estado actual. El parámetro ε se calcula como la distancia que hace que $S \subseteq \mathbb{R}^4$ tenga un número de puntos N entre $N_{\min} \leq N \leq N_{\max}$. Así pues, en nuestro experimento se establecen $N_{\min} = 200$ y $N_{\max} = 1000$.

En resumen:
$$\min ID_i \quad \text{s.a} \quad N_{\min} \leq N \leq N_{\max} \quad (5.3)$$

$$d_i \leq \varepsilon \quad \forall i = 1, 2, \dots, N$$

El criterio de decisión que utiliza este algoritmo es exclusivamente el índice de desempeño asegurando que haya un número determinado de puntos en el conjunto S . Podemos decir que al no incluir la distancia como variable de decisión el controlador tomará siempre el valor de los actuadores con menor ID, pero es muy probable que no esté cerca del punto de trabajo y lleve a nuestro sistema a otro punto no deseado. Este problema se intenta reducir al restringir los puntos entre N_{\min} y N_{\max} haciendo que el conjunto de puntos que analiza no estén muy alejados y sea una muestra suficiente.

Otro paso importante para obtener las trayectorias candidatas es elegir ε .

Para determinar ε se utiliza el siguiente algoritmo. En primer lugar, resolvemos un problema de aproximación (fig 5.5) haciendo ε el doble cuando $N < N_{\min}$ o la mitad cuando $N > N_{\max}$. No obstante para obligar a su vez que $N_{\min} \leq N \leq N_{\max}$ debemos aplicar además otro método para el caso de que dos ε consecutivos sean uno $N_1 < N_{\min}$ y otro $N_2 > N_{\max}$. Este segundo método es el de la bisectriz que divide aquella ε mayor cuando $N > N_{\max}$ y aquella ε menor al ser $N < N_{\min}$, en la figura 5.6 A y B respectivamente, de manera iterativa hasta que $N_{\min} \leq N \leq N_{\max}$.

También cabe la posibilidad en el método de las aproximaciones $\varepsilon \geq \varepsilon_{\max}$, es decir, buscaría puntos para trayectorias candidatas “i” demasiado alejado de su estado actual. En nuestro sistema hemos establecido $\varepsilon_{\max} = 0.64$, ya que $h = [0, 1.4]$. En ese caso se opta por tomar el valor de los actuadores con menor índice de desempeño. Esto se da sólo cuando los tanques parten de alturas muy diferentes, esto es, cuando dos de los tanques se encuentran totalmente llenos y los restantes vacíos.

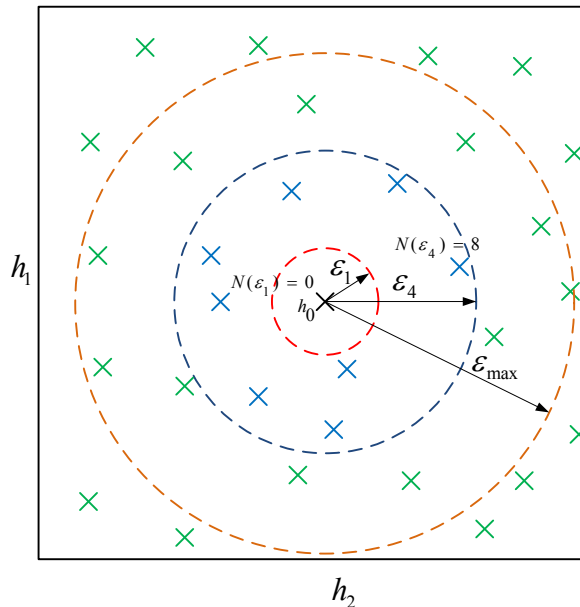


Fig 5.5. Método de aproximaciones

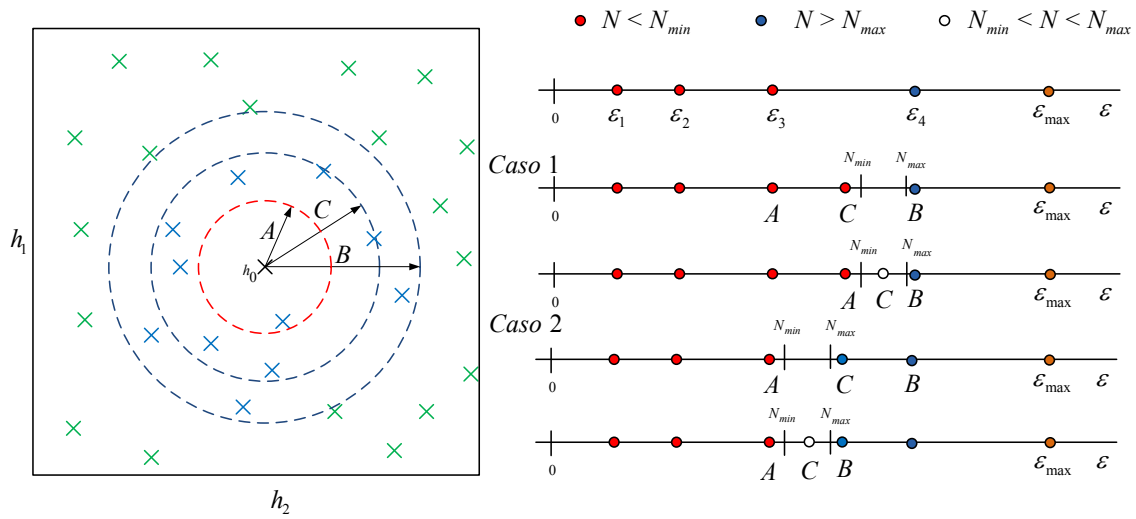


Fig 5.6. Método de la bisección

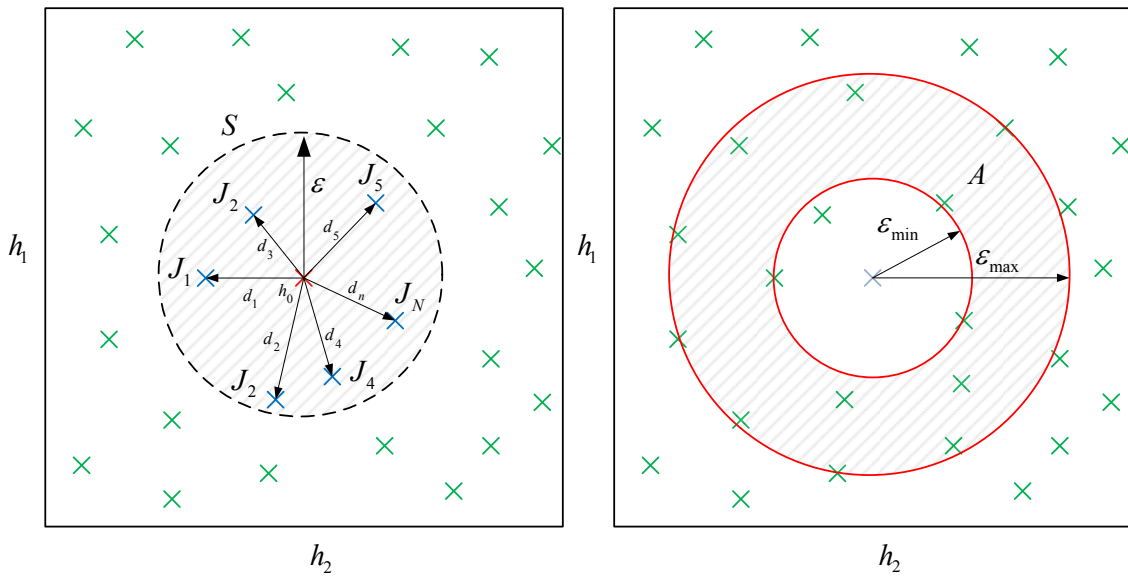


Fig 5.7. Algoritmo de búsqueda del mínimo ID

En la figura 5.7 se muestra el conjunto S que $J_1, \dots, J_N \in S$ (J_i Costes asociados a las trayectorias candidatas “i”) donde $d_i \leq \epsilon$ siendo $\epsilon_{min} \leq \epsilon \leq \epsilon_{max}$. Para aclarar esto se indica que $\epsilon \in A$. Cabe decir que estos parámetros cambian dependiendo de la densidad de puntos de la base de datos que estemos tratando.

Para llevar a cabo la búsqueda de estados iniciales próximos se puede optar entre varias posibilidad como la norma infinito fig 5.8.a y 5.8.b la norma euclídea. En nuestro caso se ha optado por norma euclídea.

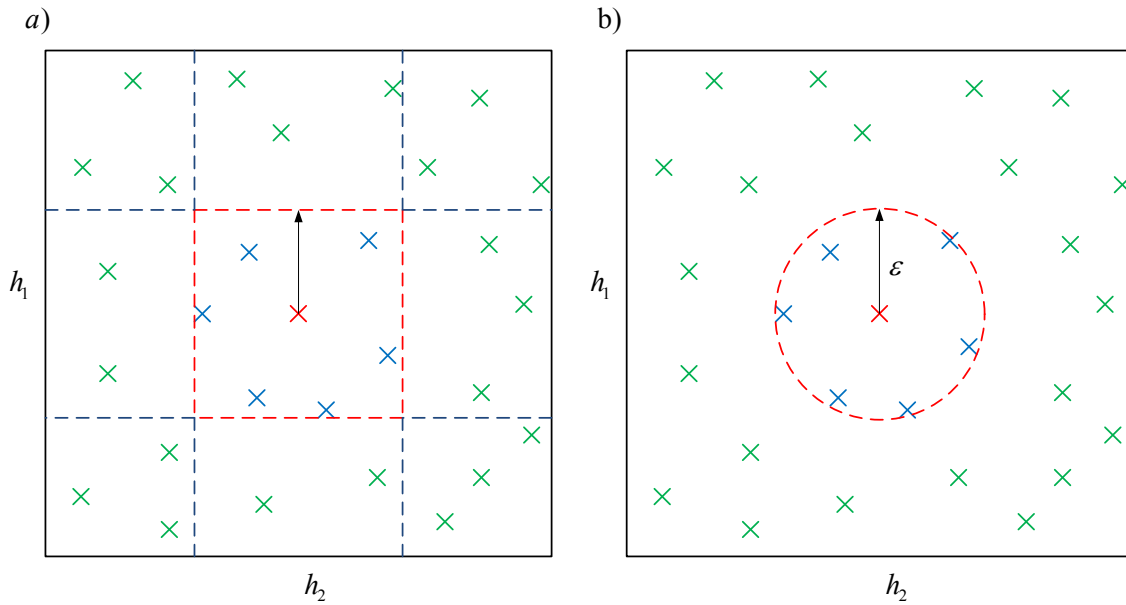


Fig 5.8. Diferentes modos de búsqueda de estados iniciales

Hay que señalar que para la obtención de los estados iniciales próximos a nuestro estado actual en nuestro algoritmo se toman varias restricciones que condicionan el funcionamiento de dicho controlador. Por un lado, hay asociado un único estado inicial a cada trayectoria, ya que si no hacemos esta restricción se puede llegar al caso que aun teniendo $N_{\min} \leq N \leq N_{\max}$ siendo $N_{\min} = 200$ y $N_{\max} = 1000$, en realidad, se analicen únicamente tres o cuatro trayectorias diferentes. Por otro lado, se analizan exclusivamente aquellas trayectorias pertenecientes a los estados iniciales cuyo horizonte de predicción no sobrepase el tiempo total de cada simulación almacenada desde el punto de trabajo actual.

Por último, recalcar que sólo se calcula el ID sobre una única trayectoria, en otras palabras, una vez se escoja el h_i , se aplica la acción de control de esa trayectoria y se descartan las demás.

$$q_k = q_{i_k} \quad \forall k = 1, 2 \quad (5.4)$$

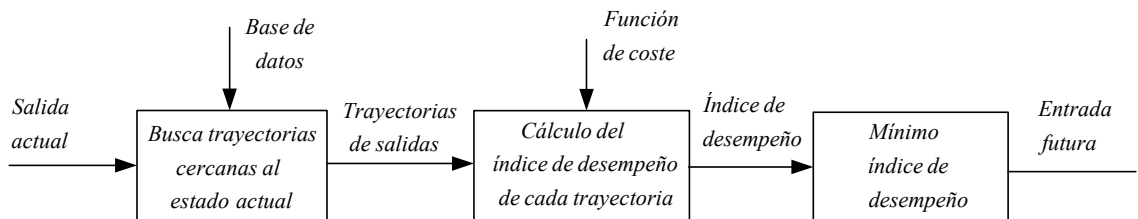


Fig 5.9. Esquema del control predictivo basado en base de datos con método de mínimo ID

5.3.2 Método de índices ponderados

Este método está basado en el del mínimo desempeño salvo que no toma el valor del actuadores del punto con menor ID, sino de los cuatro puntos con menor ID aplicando una ponderación que penaliza más a los puntos con mayor coste y pondera de modo ventajoso a aquel con menor ID. Así pues se llega a que:

$$\min ID_i \quad \text{s.a} \quad N_{\min} \leq N \leq N_{\max} \quad (5.5)$$

$$d_i \leq \varepsilon \quad \forall i = 1, 2 \dots N$$

De estas 4 trayectorias con menos índice de desempeño se hace una media ponderada en función de su coste se tiene una solución explícita que se llega de la siguiente forma:

$$\text{Donde} \quad \sum_{i=1}^{n=4} \alpha_i = 1 \quad (5.6)$$

$$q_k = \alpha_1 (ID_{\min_1}) \cdot q_k + \alpha_2 (ID_{\min_2}) \cdot q_k + \alpha_3 (ID_{\min_3}) \cdot q_k + \alpha_4 (ID_{\min_4}) \cdot q_k \quad \forall k = 1, 2 \quad (5.7)$$

$$\alpha_k = \frac{2}{N} - \frac{ID_i}{\sum_{i=1}^N ID_i} \quad k = 1, 2, 3, 4 \quad (5.8)$$

Con este método se intenta mitigar algunos problemas que presentaba el anterior controlador, ya que a veces oscilan mucho los valores del actuador al encontrarse dos puntos óptimos consecutivos alejados y cercanos respectivamente del punto de trabajo. De esta forma la respuesta se suaviza, aunque no mejora en gran medida a su precedente.

5.3.3 Método mediante uso de programación lineal

La programación lineal es un procedimiento o algoritmo matemático mediante el cual se resuelve un problema indeterminado, formulado a través de un sistema de inecuaciones lineales, optimizando la función objetivo, también lineal. Consiste en optimizar (minimizar o maximizar) una función lineal, en nuestro caso minimizar, denominada función objetivo, de tal forma que las variables de dicha función estén sujetas a una serie de restricciones que expresamos mediante un sistema de inecuaciones lineales. [5]. En nuestro caso haremos uso de la programación lineal para obtener una combinación convexa de los estados iniciales para obtener un menor coste medio.

Nuestro objetivo es minimizar el coste medio obtenido a partir de la combinación de los costes de las trayectorias incluidas. Para ello se fuerza la combinación convexa con las siguientes restricciones. Éstas se formulan de la siguiente forma:

$$\min \sum_{i=1}^N \alpha_i ID_i \quad \text{s.a} \quad \sum_{i=1}^N \alpha_i = 1 \quad (5.9)$$

$$\sum_{i=1}^N \alpha_i h_i = h_0 \quad (5.10)$$

$$\alpha_i \geq 0 \quad \forall i = 1, 2, \dots, N \quad \text{siendo } h_i, h_0 \in \mathbb{R}^4 \quad (5.11)$$

$$d_i \leq \varepsilon \quad \forall i = 1, 2, \dots, N \quad d_i = \|h_i - h_0\| \quad (5.12)$$

Así como para el método del mínimo índice de desempeño se restringe que los estados iniciales próximos que se estudian disten como máximo ε del estado actual. También decir dependiendo de ε tendremos mayor o menor número de estados iniciales bajo estudio (N). No obstante, ambos parámetros dependerán de la factibilidad o no factibilidad del problema de optimización. En definitiva, en este caso para obtener dichos parámetros se implementará otro algoritmo que se explicará a continuación.

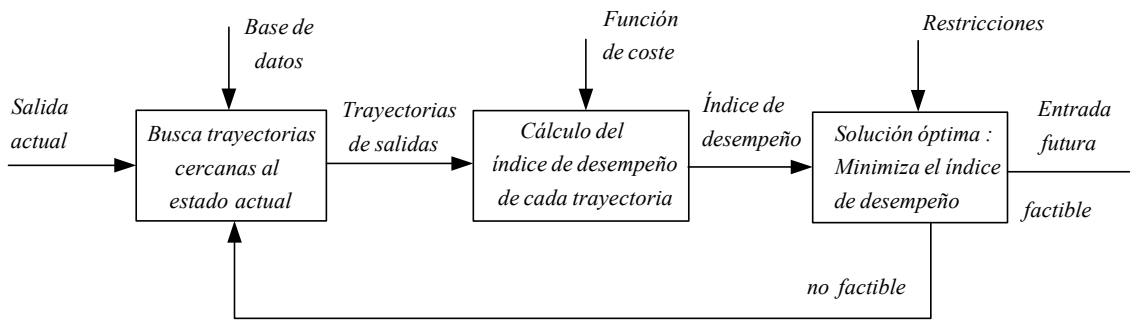


Fig 5.10. Esquema de control predictivo basado en base de datos usando programación lineal

Para resolver el problema de minimizar el coste medio con restricciones utilizaremos la función linprog de las librerías de matlab. Esta función nos proporciona el mínimo de un problema que especificamos de la forma:

$$\min f^T x \quad \text{tal que} \quad \begin{cases} A \cdot x \leq b, \\ A_{eq} \cdot x = b_{eq}, \\ lb \leq x \leq ub. \end{cases} \quad (5.13)$$

Así pues, se llama a la función enviándole el siguiente problema, formulado de la siguiente forma:

$$\text{Siendo} \quad \sum_{i=1}^N \alpha_i h_i = h_0 \quad \text{para } i = 1, \dots, N \quad \text{siendo } h_i, h_0 \in \mathbb{R}^4 \quad (5.14)$$

$$\begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ h_1 & h_2 & \cdots & h_{N-1} & h_N \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} 1 \\ h_0 \end{pmatrix} \quad (5.15)$$

$$A_{eq} = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ h_1 & h_2 & \cdots & h_{N-1} & h_N \end{pmatrix} \quad b_{eq} = \begin{pmatrix} 1 \\ h_0 \end{pmatrix} \quad x = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix} \quad f = \begin{pmatrix} J_1 \\ \vdots \\ J_N \end{pmatrix} \quad lb = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad ub = \begin{pmatrix} Inf \\ \vdots \\ Inf \end{pmatrix}$$

Para que nuestro problema tenga solución, la función bajo estudio f ha de ser convexa. esto es. que tenga un mínimo. En nuestro problema esa restricción se reduce a que $h_0 \in S$, que h_0 esté dentro de la envoltura convexa de los N puntos iniciales de las trayectorias candidatas.

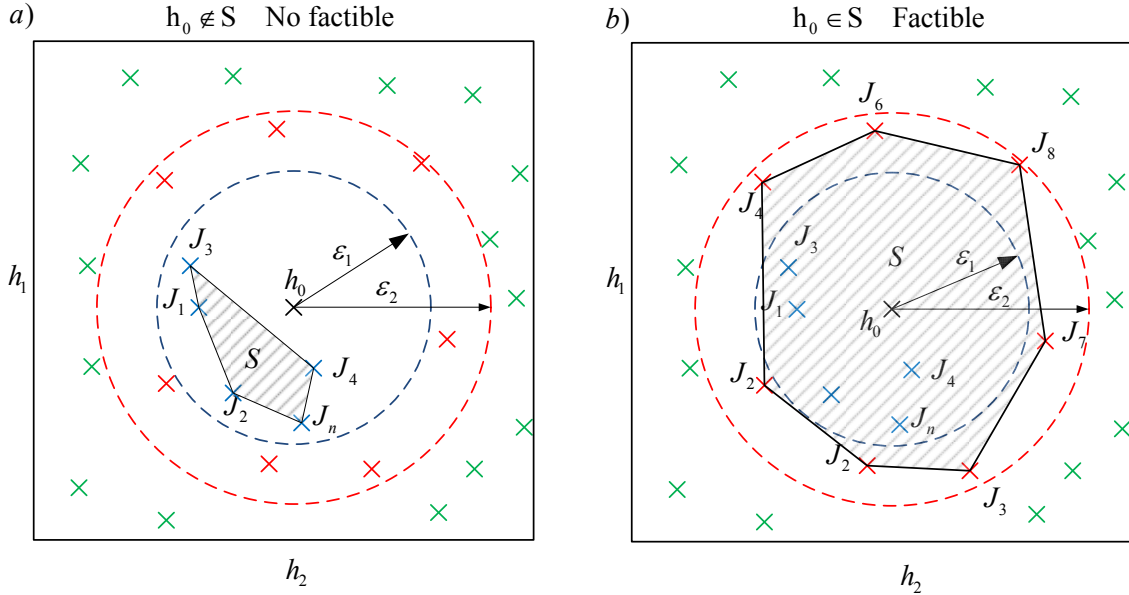


Fig 5.11. Estudio de la factibilidad del problema de optimización

En la figura 5.11.a se muestra como para $\varepsilon = \varepsilon_1$ no se encuentra una solución convexa, ya que $h_0 \notin S$. Para llegar a que el problema sea factible aumentamos hasta $\varepsilon = \varepsilon_2$ donde $h_0 \in S$ (figura 5.11.b).

Para determinar ε se utiliza el siguiente algoritmo. En primer lugar, resolvemos un problema de aproximación haciendo ε el doble cuando el problema de optimización no sea factible. Esto hace que se converja a una solución de forma rápida, no obstante no es la solución con el mínimo número de puntos y, por tanto, hace que la carga computacional no sea la óptima. Por ello tratamos de obtener una solución óptima aplicando el método de la bisección para hallar aquella solución factible con menor número de puntos. El método de la bisección divide aquella ε mayor no factible y aquella ε menor factible, en la figura 5.13 A y B respectivamente, de manera iterativa hasta que $B - C \leq Tol$. Además el método de las aproximaciones también se aplica cuando se parte

de una solución convexa de la misma forma que partiendo de una no convexa, sólo que esta vez se divide ε entre 2, aplicándose luego el método de la bisectriz.

También cabe la posibilidad en el método de las aproximaciones $\varepsilon \geq \varepsilon_{\max}$, es decir, buscaría estados iniciales candidatos para la convexidad demasiado alejado de su estado actual. En nuestro sistema hemos establecido $\varepsilon_{\max} = 0.64$, ya que $h = [0, 1.4]$. En ese caso donde no se encuentra una solución factible se opta por tomar el valor de los actuadores del punto más cercano al estado actual. Esto se da sólo cuando los tanques parten de alturas muy diferentes, esto es, cuando dos de los tanques se encuentran totalmente llenos y los restantes vacíos.

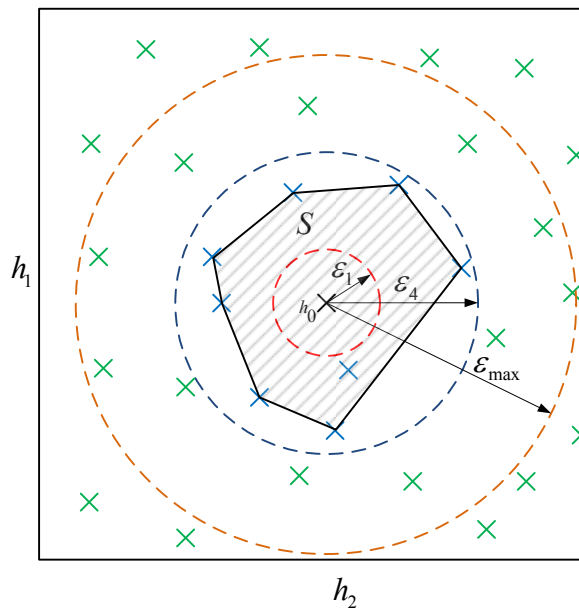


Fig 5.12. Método de aproximaciones

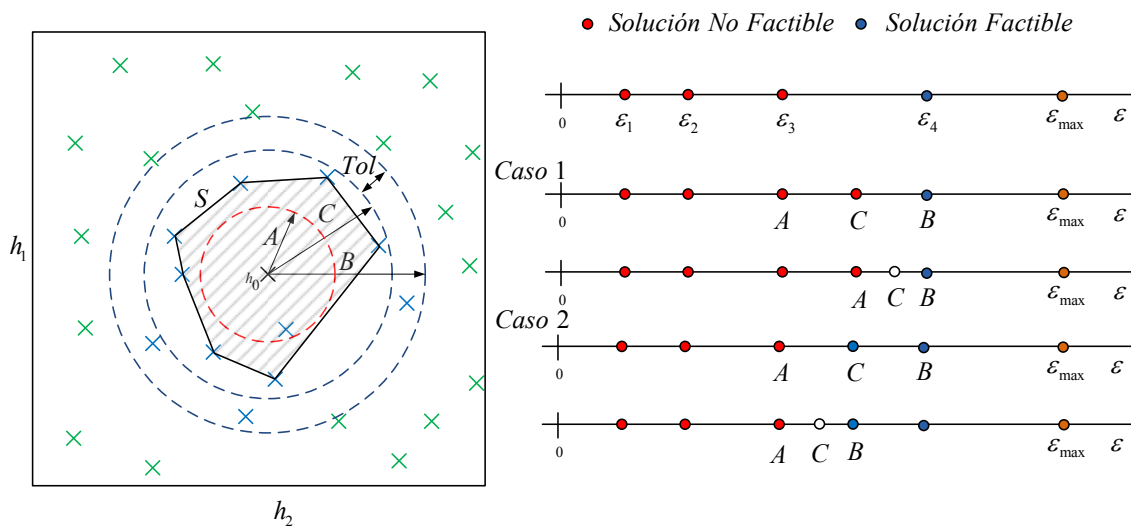


Fig 5.13. Método de la bisectriz

ε_0 y Tol se han calculado de manera experimental para que el número de iteraciones en el caso de la aproximación N_{iter}^{aprox} y de la bisección N_{iter}^{bisecc} no sea demasiado grande para que realmente reduzca el coste computacional. El número de iteraciones se ve afectado tanto por ε_0 como por Tol . Por una parte, si nos acercamos o nos alejamos mucho de ε cuando definimos ε_0 hace que N_{iter}^{aprox} aumenten. Por otra parte, el número de iteraciones en el método de la bisección aumenta sólo si se disminuye Tol , siendo $N_{iter}^{bisecc} = 0$ si Tol es muy grande. Por consiguiente, se opta por una solución de compromiso entre el número de iteraciones totales $N_{iter}^{total} = N_{iter}^{aprox} + N_{iter}^{bisecc}$ y el coste computacional del cálculo con un menor número de iteraciones, pero por tanto un coste mayor al calcular un mayor número de trayectorias iniciales candidatas.

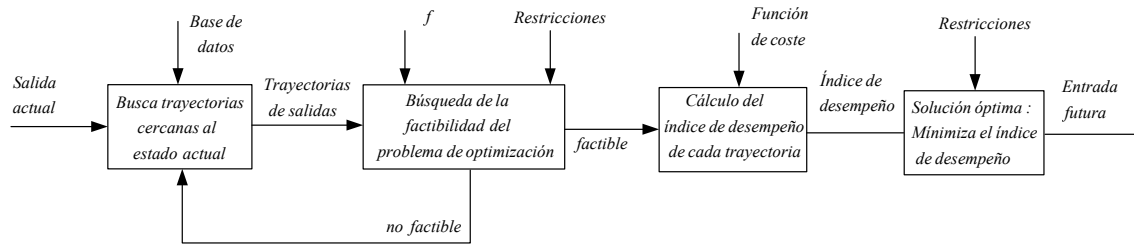


Fig 5.14. Esquema de control predictivo basado en base de datos método usando programación lineal optimizado

Una vez que hemos explicado que existen un cierto número de iteraciones N_{iter}^{total} antes de escoger ε optamos por cambiar parte del esquema del algoritmo de control para reducir el tiempo computacional.

A diferencia de la fig 5.10. en la fig 5.14. se muestra que el cálculo de los índices de desempeño sólo se calculan una vez que el problema de optimización sea factible y además $B - C \leq Tol$. Sin embargo, la función linprog necesita para su cálculo $f = \begin{pmatrix} J_1 \\ \vdots \\ J_N \end{pmatrix}$

, por lo que hacemos que $f = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ mientras tanto. Una vez se encuentre una solución factible y $B - C \leq Tol$ con la función calculoIDs.m calculamos J_1, \dots, J_N .

Cabe resaltar que a diferencia con el método del mínimo índice de desempeño o el de los índices ponderados, la distancia de los puntos al punto de trabajo actual es una variable de decisión, ya que $\sum_{i=1}^N \alpha_i h_i = h_0$ es una restricción que imponemos a nuestro problema lineal. Esto hace que ponderen α_i dependiendo de su índice de desempeño y la distancia de éstos al punto de actual.

El valor de los actuadores quedaría como:

$$\begin{aligned}
q_k = & \alpha_1(J_1, d_1) \cdot q_{k1} + \alpha_2(J_2, d_2) \cdot q_{k2} + \dots \\
& + \alpha_{N-1}(J_{N-1}, d_{N-1}) \cdot q_{k(N-1)} + \alpha_N(J_N, d_N) \cdot q_{k \cdot N} \quad \forall k = 1, 2
\end{aligned} \tag{5.16}$$

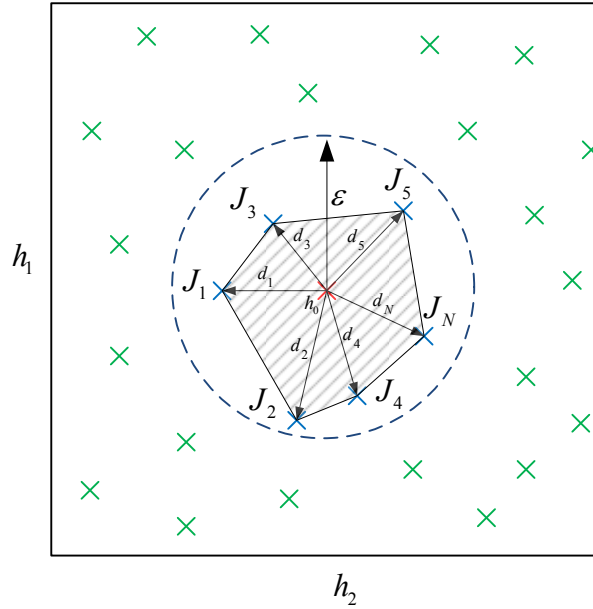


Fig 5.15. Variables de decisión del controlador predictivo por programación lineal

Hay que añadir que para comprobar si es o no factible la solución a nuestro problema es necesario tener al menos 4 puntos al buscar una convexidad de una función en \mathbb{R}^4 . Así pues nuestro algoritmo establece un $N_{\min}=4$ para posteriormente comprobar la convexidad de la función. Podemos decir que al aumentar N_{\min} la respuesta del controlador es más suave.

6 Análisis/validación del controlador

6.1 Comparativa de los índices de desempeño

6.1.1 1 trayectoria de ejemplo

En este análisis se compara el comportamiento que presenta el DbPC frente a varios controladores (P, PI, LQR). Hay que señalar que aunque los controladores P, PI y LQR son los mismos que se han usado para la creación de la base de datos, en las simulaciones no se hace uso de la base de datos para obtener la trayectoria de éstos. Así pues se utiliza el esquema de simulink (véase fig. x), que se utilizó para la creación de la base de datos, para la obtención de la trayectorias de los controladores P, PI y LQR.

En primer lugar, se realiza una simulación de 1200 segundos usando 4 de controladores estudiados partiendo desde el estado inicial $h_0 \in \mathbb{R}^4$ y hasta alcanzar la referencia $h_{ref} \in \mathbb{R}^4$.

$$\begin{aligned} h_{10} &= 0.7293 & h_{1ref} &= 0.5955 \\ h_{20} &= 0.8102 & h_{2ref} &= 0.6616 \\ h_{30} &= 0.6594 & h_{3ref} &= 0.5384 \\ h_{40} &= 0.9408 & h_{4ref} &= 0.7682 \end{aligned} \quad (6.1)$$

Así pues se representan las diferentes variables que hacen que el ID sea mayor o menor en las figuras 6.1 y 6.2. Debemos saber que tanto c , p , A y V_{min} son constantes siendo $c = 1$, $p = 20$ en nuestra trayectoria y $A = 0.03$ y $V_{min} = 0.012$ en la planta de los 4 tanques. De esta forma la bondad de nuestros controladores dependerá exclusivamente de q_a, q_b, h_1 y h_2

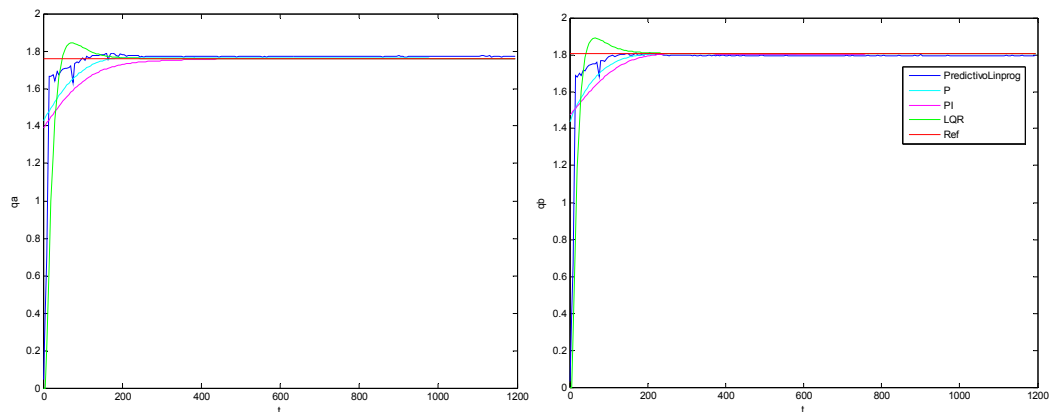


Fig 6.1. Comparativa del valor de los actuadores q_a y q_b de los diferentes controladores sobre una trayectoria

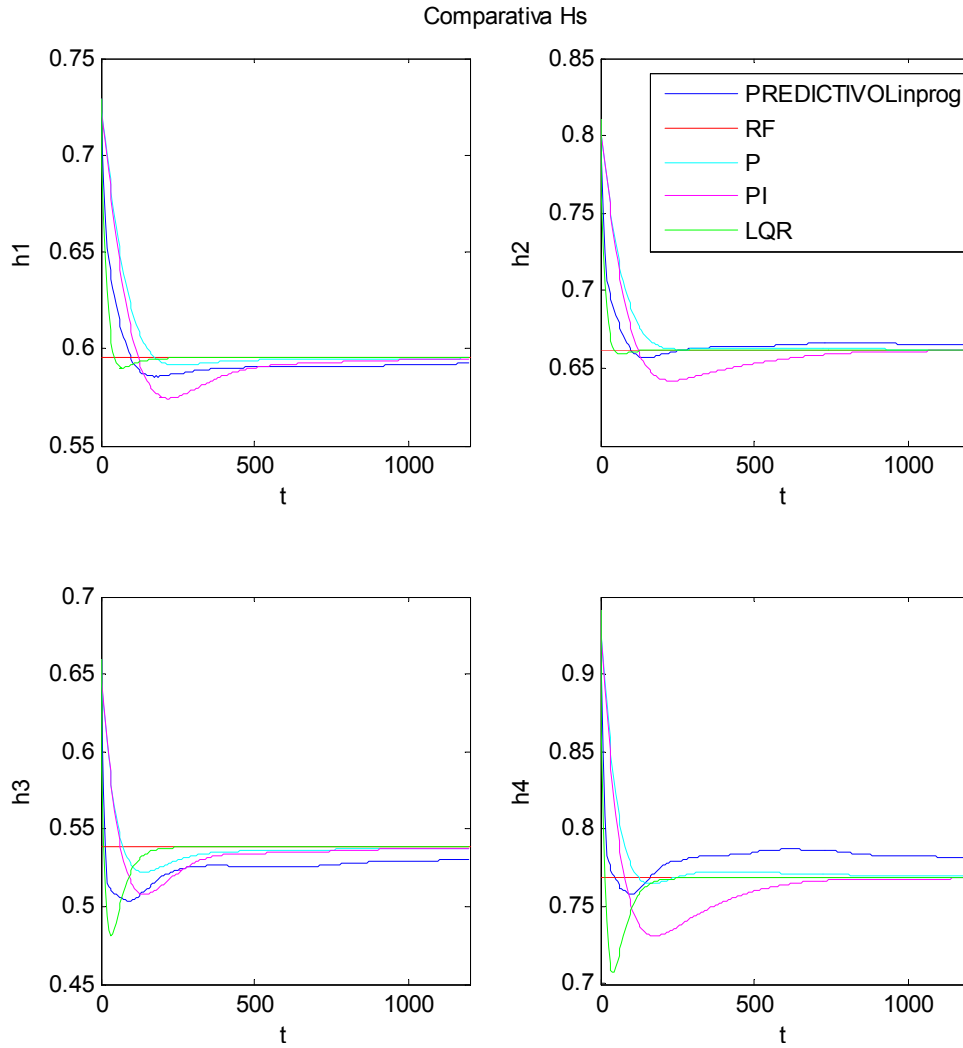


Fig 6.2. Comparativa de alturas de los diferentes controladores sobre una trayectoria

No obstante para conocer la bondad de los controladores es necesario representar una comparativa del índice de desempeño (*ID*). Así pues en la figura 6.3 se muestra el índice de desempeño de una trayectoria de los diferentes controladores que controlan la planta de los 4 tanques en bucle cerrado. El *ID* se formula de la siguiente forma:

$$ID(t) = \int_0^t (J_i(h(\tau), q(\tau), c(\tau), p(\tau)) - J^*(c(\tau), p(\tau))) d\tau \quad (6.2)$$

Donde J_i es el coste de etapa para el caso del problema de los cuatro tanques y se expresa de la forma:

$$J_i(h, q, c, p) = q_a^2 + c \cdot q_b^2 + p \frac{V_{\min}}{A(h_1 + h_2)} \quad (6.3)$$

Asimismo J_i que se va a considerar consta de dos términos asociados a los objetivos principales:

J_{ce} es el coste de operación de la minimización del consumo energético de la planta y

J_{va} es el coste de operación del volumen acumulado en los depósitos inferiores.

$$J_i = J_{ce} + J_{va} \quad (6.4)$$

$$J_{va} = p \frac{V_{\min}}{A(h_1 + h_2)} \quad (6.5)$$

$$J_{ce} = q_a^2 + c \cdot q_b^2 \quad (6.6)$$

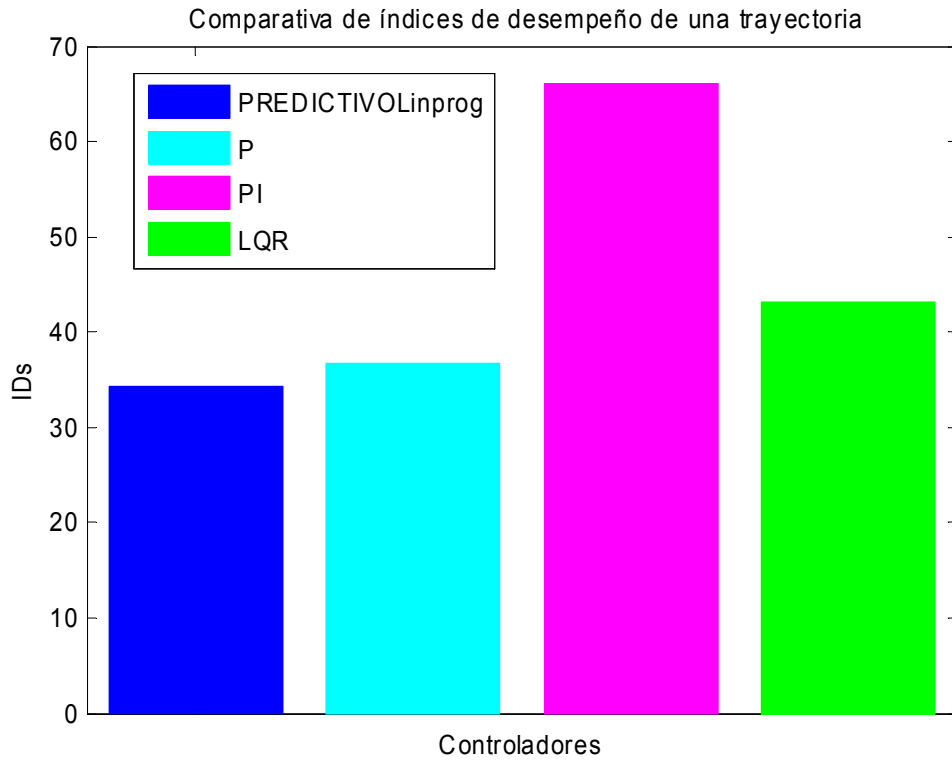


Fig 6.3. Comparativa del índice de desempeño de los controladores estudiados sobre una trayectoria

En esta figura 6.3 podemos observar como el DbPC se comporta mejor que los demás controladores, ya que presenta un menor ID . También se añade la figura 6.4 para mostrar el índice de desempeño asociado a la minimización del consumo energético ID_{ce} y por otra parte a la maximización del volumen acumulado ID_{va} . Esta última gráfica nos ayuda a entender el valor de la minimización de la función de coste, ya que a pesar de que el DbPC no tenga el menor ID de los dos términos asociados a los objetivos (ID_{va} y ID_{ce}) en comparación con el resto de controladores, posee el menor ID de la suma de estos términos.

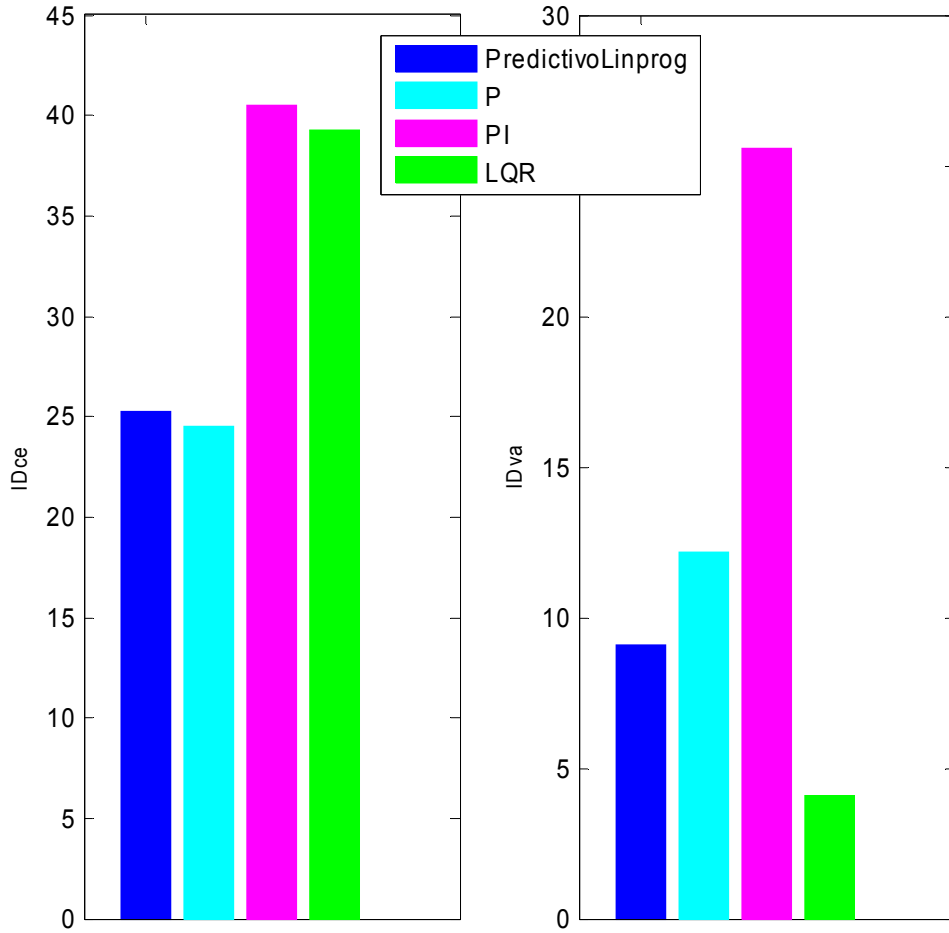


Fig 6.4. Comparativa del valor de los actuadores q_a y q_b de los diferentes controladores sobre una trayectoria

Cabe señalar que en las figuras anteriores se ha comparado el DbPC usando programación lineal frente a los diferentes controladores P, PI y LQR. No obstante, hemos implementado otros métodos basados en el DbPC, aunque menos eficaces también realizables. Estos eran:

- Método de mínimo índice de desempeño
- Método de índices ponderados
- Método mediante uso de programación lineal

En la figura 6.5 se muestra la comparativa de las alturas de los diferentes métodos realizando una simulación de mil doscientos segundos usando los tres controladores antes mencionados partiendo desde el estado inicial $h_0 \in \mathbb{R}^4$ y hasta alcanzar la referencia $h_{ref} \in \mathbb{R}^4$.

$$\begin{aligned}
 h_{10} &= 0.4675 & h_{1ref} &= 0.5955 \\
 h_{20} &= 0.6793 & h_{2ref} &= 0.6616 \\
 h_{30} &= 0.3807 & h_{3ref} &= 0.5384 \\
 h_{40} &= 0.5432 & h_{4ref} &= 0.7682
 \end{aligned} \tag{6.7}$$

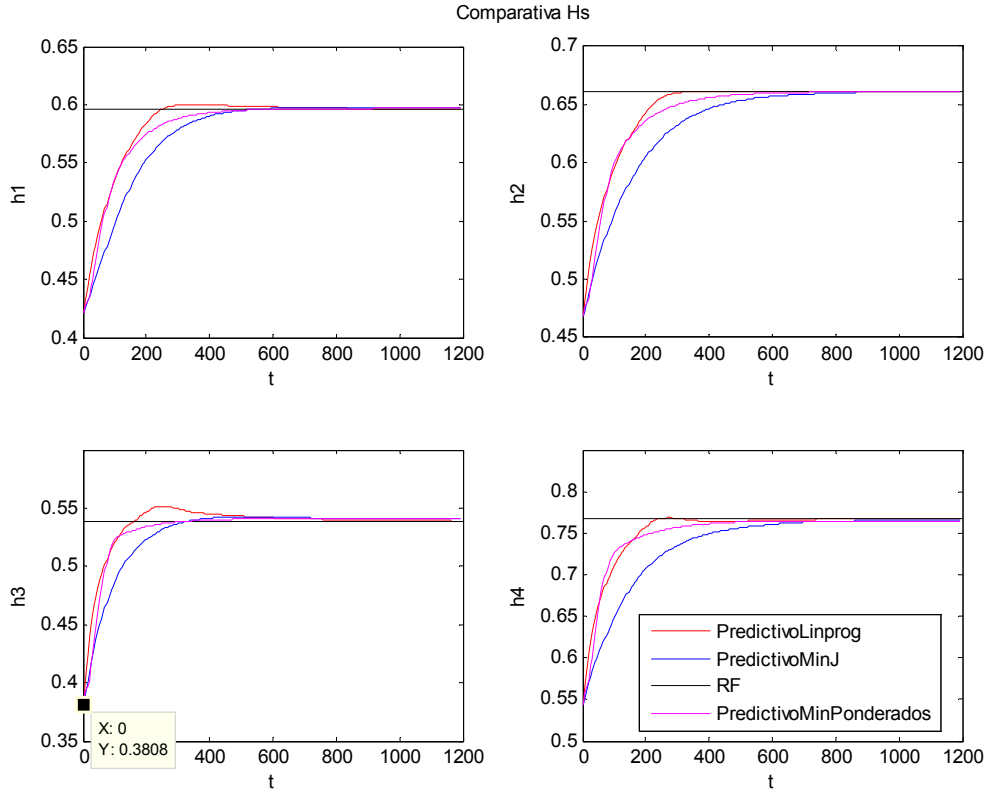


Fig 6.5. Comparativa de alturas de los diferentes controladores predictivos sobre una trayectoria

6.1.2 100 trayectorias aleatorias. Valores medios

En este apartado se muestra una comparativa de los controlares DbPC (usando programación lineal), P, PI y LQR. Para ello, en este caso, se simulan cien conjuntos de trayectorias que parten de un estado inicial $h_0 \in \mathbb{R}^4$ aleatorio, siendo $h_0 = \{h_{01}, h_{02}, h_{03}, h_{04}\}$, y queremos llevar el sistema al punto de operación que hemos definido como $h_{ref} \in \mathbb{R}^4$ siendo $h_{ref} = \{h_{ref_1}, h_{ref_2}, h_{ref_3}, h_{ref_4}\}$ y $q_a^*, q_b^* \in \mathbb{R}$. Por tanto hacemos llevar al sistema en cada conjunto de trayectorias desde cada h_0 hasta h_{ref} mediante los cuatro controladores antes mencionados.

La media del *ID* de los cien conjunto de trayectorias que se muestran en la figura 6.6 permitiéndonos concluir que el controlador DbPC tiene menor *ID*.

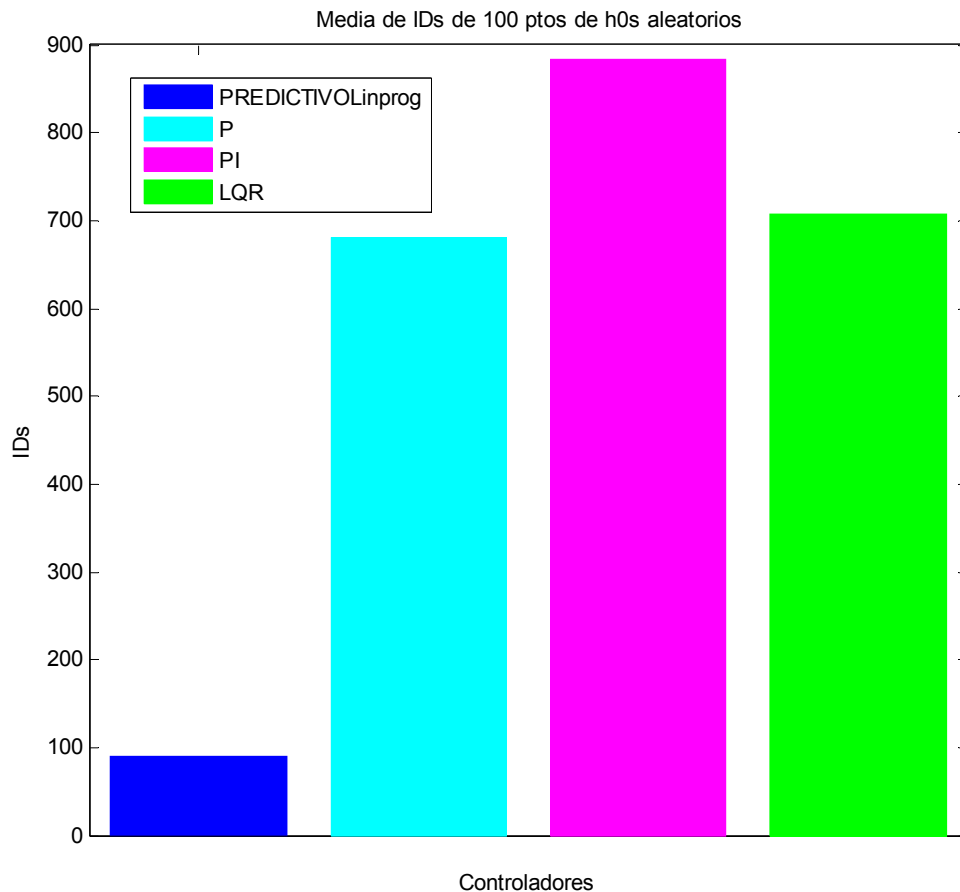


Fig 6.6. Comparativa de ID de los diferentes controladores en 100 conjunto de trayectorias partiendo de h_0 aleatorias.

6.2 Optimización de la base de datos

6.2.1 Estimación de M óptimo

Una de las incógnitas que se nos presenta en la programación de un DbPC es el tamaño de las bases de datos. Esto es un triple problema. Por un lado, existe el problema del espacio en memoria, ya que tomando como ejemplo las bases de datos utilizadas en el DbPC dependiendo del número de trayectorias almacenadas en la base de datos oscila entre los trescientos megabytes para una base de datos de quince mil trayectorias o treinta megabytes para una de trescientas trayectorias. Por otro lado, en nuestra implementación al aplicar base de datos con diferente número de trayectorias influye en el tiempo computacional. Y por último, tiene influencia sobre el *ID*.

De esta forma el objetivo será buscar el número óptimo de trayectorias que debemos almacenar en nuestra base de datos para poder controlar nuestro sistema y a su vez tener un *ID* bajo.

Para la búsqueda de la estimación de dicho número óptimo de trayectorias se aborda el problema de la influencia del *ID*. Se opta por calcular el *ID* de cincuenta y cinco

trayectorias haciendo uso de diferentes bases de datos de diferente tamaño. Estas trayectorias parten de un estado inicial $h_0 \in \mathbb{R}^4$ aleatorio, siendo $h_0 = \{h_{01}, h_{02}, h_{03}, h_{04}\}$, y queremos llevar el sistema al punto de operación que hemos definido como $h_{ref} \in \mathbb{R}^4$ siendo $h_{ref} = \{h_{ref1}, h_{ref2}, h_{ref3}, h_{ref4}\}$ y $q_a^*, q_b^* \in \mathbb{R}$, así como indicar que todas las bases de datos convergen a una misma h_{ref} .

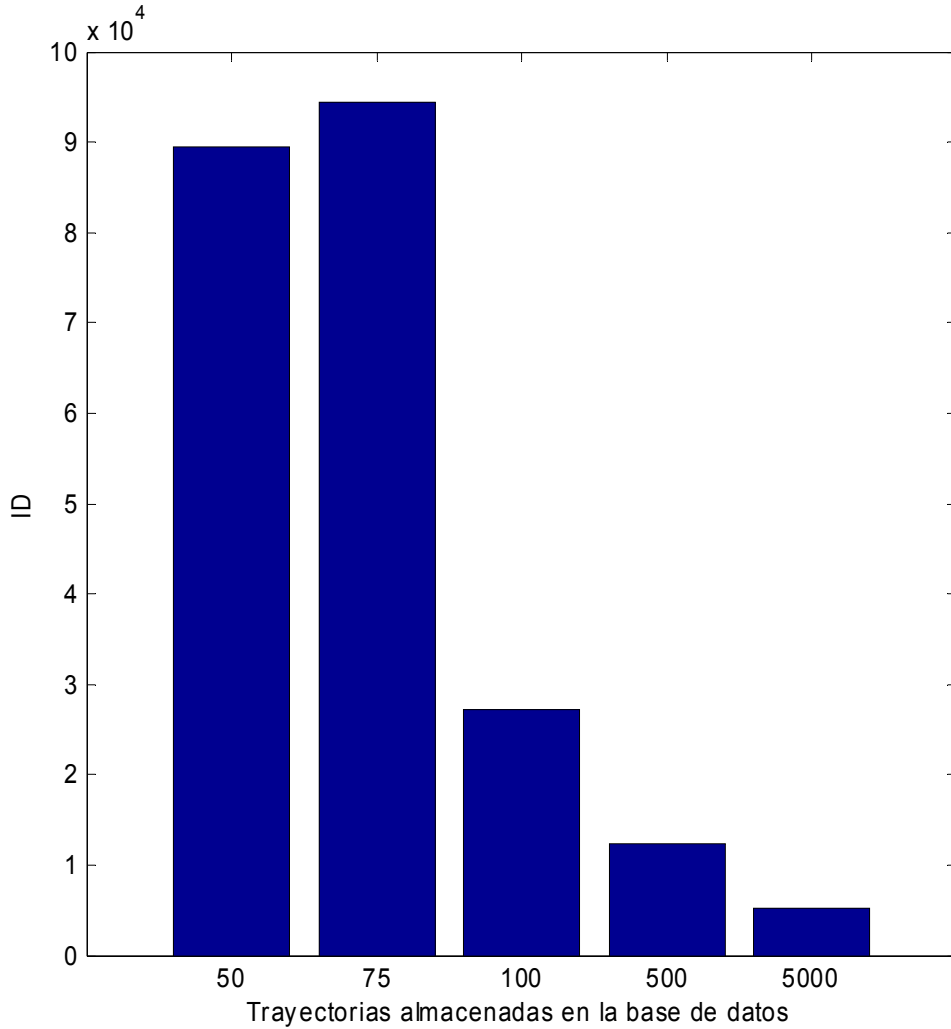


Fig 6.7. Representación de la suma de ID de un conjunto de trayectorias usando bases de datos de diferente tamaño

La figura 6.7 muestra que el número de trayectorias almacenadas en una base de datos es proporcional al ID . Sin embargo, para bases de datos a partir de mil conjuntos de trayectorias el ID es aproximadamente el mismo, así como para bases de datos con menos de cien conjuntos de trayectorias no existe una relación clara entre ID y número de trayectorias. En resumen, para resolver el problema de seguimiento de un punto de referencia en la planta de cuatro tanques debemos tomar una base de datos que almacene un conjunto de trayectorias entre cien y mil para poder controlarla con garantías y sin aumentar demasiado el coste computacional y reducir nuestro espacio de memoria.

6.2.2 Coste medio/tiempo de cálculo medio/tamaño de base de datos

Un problema común en la familia de los controladores predictivos es el coste computacional. Este hecho se pone de relieve también en el DbPC, ya que la carga de cálculo necesaria para la resolución de algunos algoritmos hace que los tiempos de simulación sean bastante elevados.

Como muestra de ello se muestra la figura 6.8 donde se presenta los costes computacionales de una misma simulación tomando diferentes bases de datos de diferente tamaño. Al igual que en apartado 6.1 se realiza una simulación de 1200 segundos usando 4 de controladores estudiados partiendo desde el estado inicial $h_0 \in \mathbb{R}^4$ y hasta alcanzar la referencia $h_{ref} \in \mathbb{R}^4$.

$$\begin{aligned}
 h_{10} &= 0.7293 & h_{1ref} &= 0.5955 \\
 h_{20} &= 0.8102 & h_{2ref} &= 0.6616 \\
 h_{30} &= 0.6594 & h_{3ref} &= 0.5384 \\
 h_{40} &= 0.9408 & h_{4ref} &= 0.7682
 \end{aligned} \tag{6.8}$$

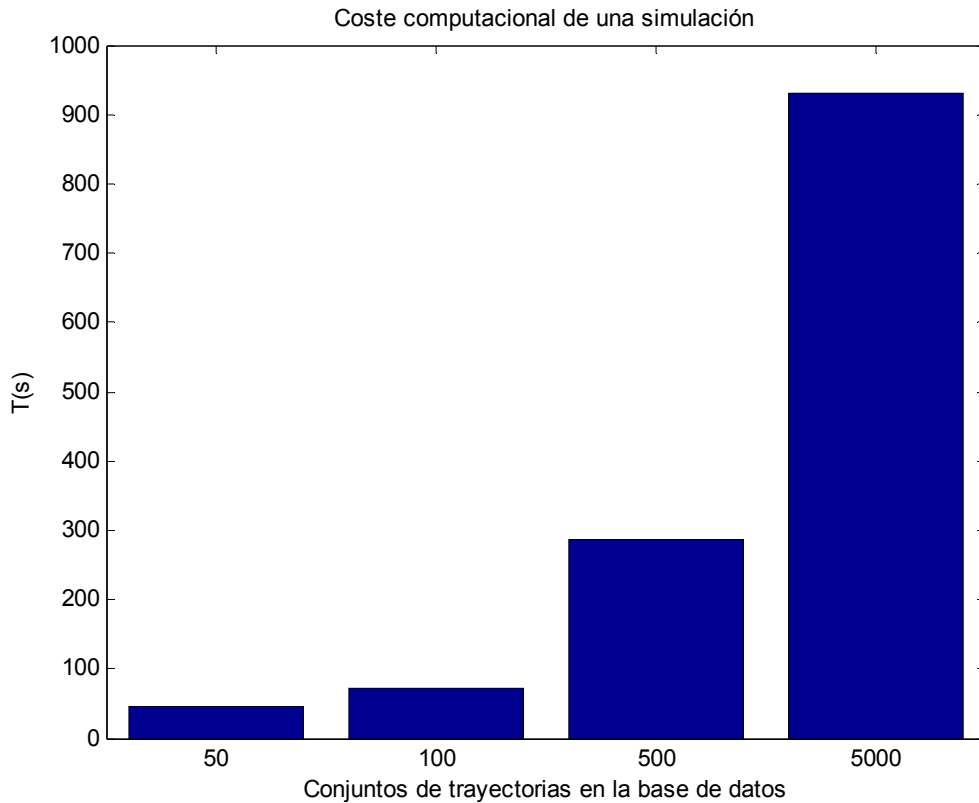


Fig 6.8 Representación del coste computacional de una simulación con bases de datos con diferente tamaño

A partir de los resultados obtenidos en las gráficas 6.7 y 6.8, donde se representa la bondad de los controladores y la carga de cálculo respectivamente, podemos concluir que hay que llegar a una solución de compromiso en la elección del tamaño de la base de datos. Así pues esta solución debe tener en cuenta que una base de datos con un gran número de trayectorias nos garantiza una solución con menor índice de desempeño, pero por el contrario nos penaliza con un coste computacional mayor.

7. Aplicación al concurso

En este apartado se aplica el controlador DbPC en el concurso de la CEA al caso 2, control multivariable. De esta forma se lleva a cabo el ensayo tipo propuesto en el concurso. Los controladores se ensayarán sobre la planta partiendo de un punto de equilibrio óptimo para un valor los parámetros $c=1$ y $p=20$.

- El ensayo comienza variando los precios unitarios que cambian a $c=1.5$ y $p=20$.
- Transcurridos 1200 segundos, los precios unitarios cambian a $c=1$ y $p=40$.
- Transcurridos 1200 segundos, los precios unitarios cambian a $c=0.5$ y $p=10$.
- Transcurridos 1200 segundos se vuelve al punto inicial correspondiente a los precios unitarios $c=1$ y $p=20$.
- Transcurridos 1200 segundos, se finaliza el ensayo

Indicar que para la simulación de este experimento se han tenido que generar cuatro bases de datos que convergen a los diferentes puntos objetivos que plantea el concurso.

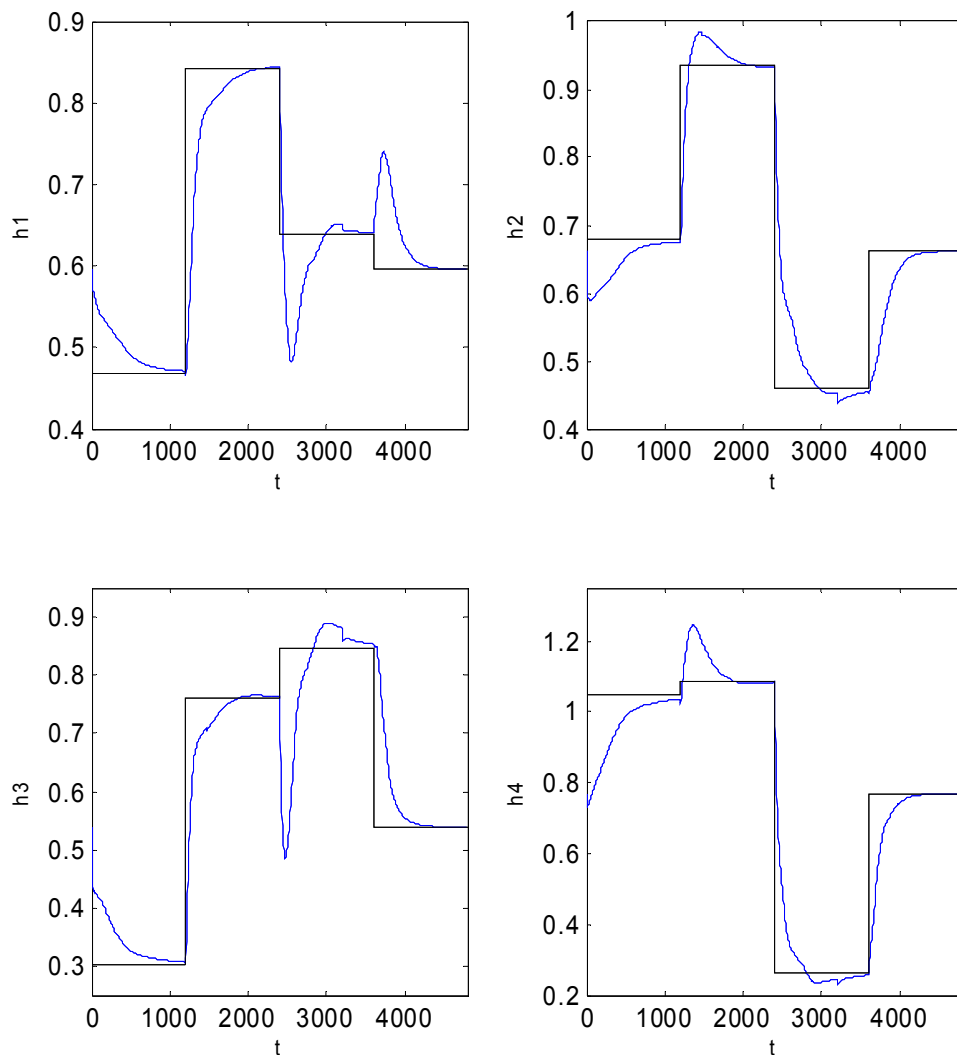


Fig 7.1. Representación de las alturas del concurso de la CEA usando un controlador DbPC

También se muestra en la figura 7.2 la comparativa de las alturas del DbPC con respecto a los controladores P, PI y LQR.

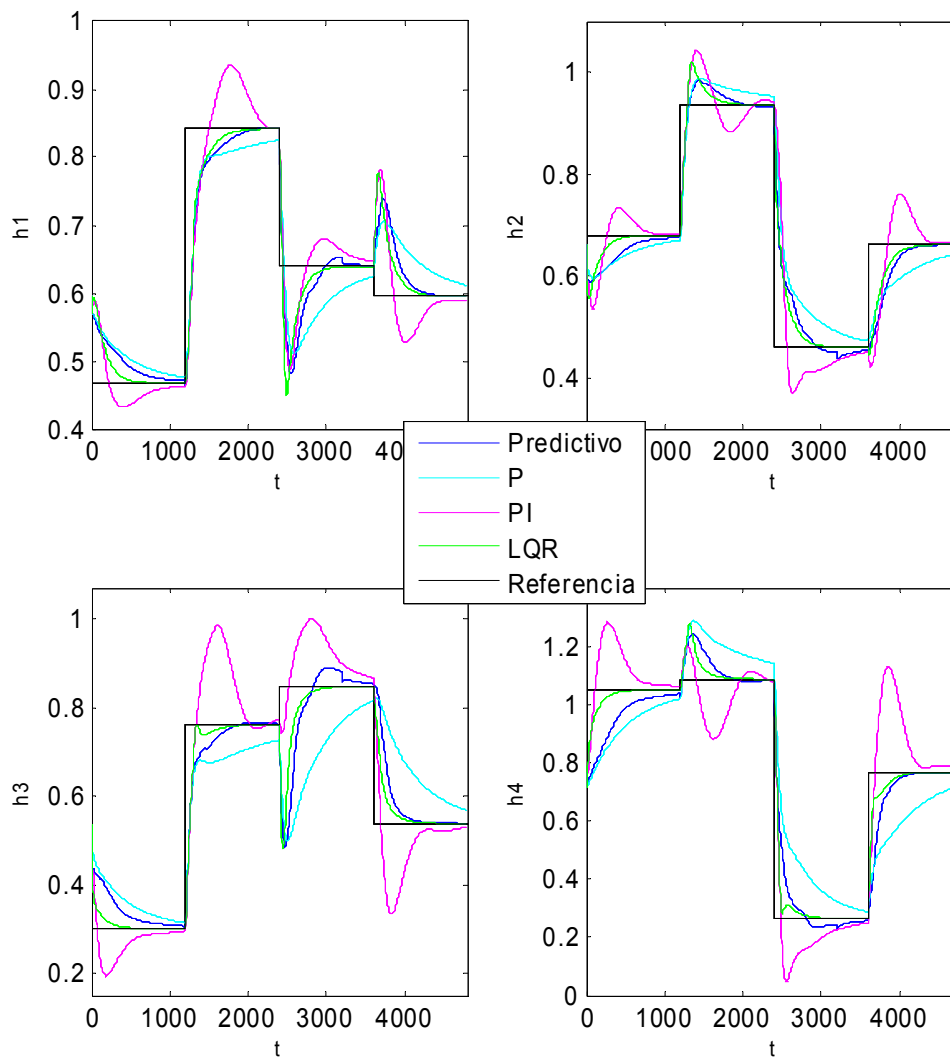


Fig 7.2. Comparativa de las alturas del concurso de la CEA usando un controlador DbPC, P, PI y LQR

Por último, para determinar la bondad del DbPC en el ensayo de la CEA se calcula el *ID* de cada controlador. Esto se representa en la figura 7.3 donde se observa que tiene un menor *ID*.

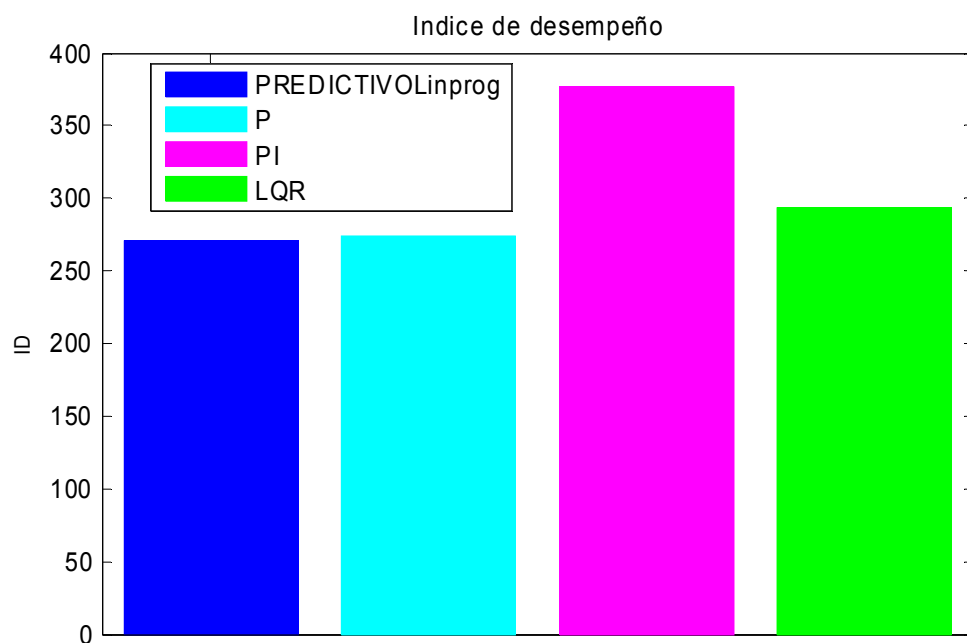


Fig 7.3. Comparativa de los índices de desempeño del concurso de la CEA usando un controlador DbPC, P, PI y LQR

8. Referencias

- [1] APUNTES DE INGENIERÍA DE CONTROL-Daniel Rodríguez Ramírez, Carlos Bordón Alba
- [2] OGATA, 98
- [CEA] Concurso de la CEA
- [4] “The quadruple-tank process” IEEE Transactions on Control Systems Technology, vol 8 (2000)
- [5] Wikipedia (Principios de la Programación Lineal)

